NetBlazer® Version 2.1 Release Notes

Table of Contents

1 Introduction

General 1
Your Comments and Questions 1
NetBlazer Update Documentation 2
System Updates 2

2 New Features/Changes

NetBlazer 2.0 Enhancements 3
NetBlazer Feature Summary by Protocol 3
IPX Support 3
AppleTalk and ARA Support 6
Enhanced SNMP Management of Modems and NetBlazer 6
Support for New Asynchronous and Synchronous Cards 7
Enhanced Multi-Line (Inverse Multiplexing) 8
Dynamic Address Assignment 8
Network Booting 8
ACS Option 8
Release 2.0 Minor Software Features 9
NetBlazer 2.1 Enhancements 9
PN Support 9
Kerberos Support on NetBlazer 9
ODI Support in PPP Shell 10
NTP 11
2.1 New Commands and Features 11
Additional Documentation 15
Version 2.1 Fixed Bugs 19

3 Precautions

Version 2.0 Precautions and Known Problems 22
Version 2.1 Precautions and Known Problems 22

4 System Update Procedures

Using Diskcopy to Update Your System 25

5 Command Reference Manual Updates

Introduction 1

Telebit is pleased to release Version 2.1 of the NetBlazer system software. Before attempting to configure and use the NetBlazer, please review this document. An overview of the new features and functions included in Version 2.1 is provided in the Enhancements section.

NetBlazer release Version 2.1 is provided on two diskettes, the System diskette, which contains 14 files, and the Utilites diskette which contains10 files. The contents of each diskette are listed in the following tables.

Table 1. System Diskette Contents

TELEBIT.SYS	BOOTP
FRED	ATK.LDR*
IPX.LDR*	SNMP.LDR
EDIT.LDR	TCL.LDR
TOKEN002.MCD	CHAT.TXT
MODEMCAP.TXT	DEFAULT.CNF
UPDATE.CNF	KRB.LDR*

Table 2. Utilities Diskette Contents

PPPCFG.EXE*	PPPCFG.HLP*
PPPCFG.MDM*	COMPPP.EXE*
IPX.COM*	WATCHPPP.EXE*
MIB.TXT	MIB.CMU
ODIPPP.COM*	NET.CFG*

Note that if you have received an IP only upgrade (a single diskette), the files marked with an asterisk (*) are not included. This diskette also includes the MIB.TXT and MIB.CMU files. For more detailed descriptions of these files, refer to the File Definitions section at the rear of the *NetBlazer Configuration Guide*.

Your Comments or Questions

If you have any comments or questions concerning this release, you can send an e-mail message to **support@telebit.com**; or call Technical Support at **1-800-TELEBIT** or **408-734-5200**.

Telebit has also established an electronic mailing list for NetBlazer users to exchange NetBlazer related questions and ideas with other NetBlazer users by sending e-mail to **netblazer-users@telebit.com**. If you want to be added to this list, send your request to **netblazer-users-request@telebit .com**.

NetBlazer Update Documentation

This release includes update sections to the existing 2.0 *NetBlazer Command Reference Manual*. These update sections are included at the end of these notes.

Future updates to the NetBlazer documentation will be available by using the FTP facility. New or updated pages to the documentation will be placed on the same server used for downloading software updates over the Internet. Since these pages are postscript files, you need a postscript printer to to take advantage of this facility.

Note: *The references to the NetBlazer Configuration Utility in Chapter 2 of the NetBlazer Instation Guides and Chapter 3 of the NetBlazer Configuration Guide are incorrect. Telebit is presently developing a configuration utility, which will be available in a future release.*

System Updates

If you are currently using an earlier version of the NetBlazer system software, you can use any of the methods described in Appendix B of the *NetBlazer Configuration Guide* to update your NetBlazer system software.

If you have any problems updating your system software, call Telebit's Technical Support at **1-800-TELEBIT** or **408-734-5200**.

New Features/Changes 2

This section overviews the new features and changes made from NetBlazer software version 1.5 to 2.0, and from version 2.0 to 2.1. Since version 2.0 was a major enhancement release while 2.1 is an incremental release, an overview of both versions is provided.

NetBlazer 2.0 Enhancements

NetBlazer software version 2.0 was designed to enhance the NetBlazer in two areas: multi-protocol support and central-site administration. In the area of multiprotocol support, the NetBlazer can support IPX, AppleTalk Phase 2 and ARA in addition to its earlier support of TCP/IP.

In the area of central-site administration, NetBlazer version 2.0 provided the following enhancements:

- Modem MIB
- Multi-Port Synchronous Cards
- Enhanced Flexibility with Asynchronous Ports
- Improved Inverse Multiplexing
- Dynamic IP Address Assignment for PPP interfaces
- Network Booting

The following chart overviews the NetBlazer 2.0 features by protocol.

NetBlazer Feature Summary by Protocol

Feature	TCP/IP	IPX	AppleTalk
LAN and WAN Routing			
Ethernet to Ethernet	✓	✓ Ethernet II and Novell 802.3 framing	✓
Ethernet to Token Ring	✓	✓	
Token Ring to Token Ring	✓	✓ Token Ring and SNAP framing	
Dial-Up Async to 115.2 Kbps	✓	✓	✓
Dial-Up Synch - Switched 56	✓	✓	✓

Feature	TCP/IP	IPX	AppleTalk
Dial-Up Synch - Basic rate ISDN	✓	✓	✓
Leased Lines to T1/E1	✓	✓	✓
Inverse Multiplexing	✓		
Tunneling over IP	✓	✓	
Header Compression	Up to 10 to 1	Up to 16 to 1	
Leased Line Backup	✓	✓	✓
Dial-In (terminal) Service & Dial-Out			
Remote Node - Dial-In access	✓	✓ Requires ACS option	✓
Dumb terminal - Dial-In access	✓	NA	NA
Remote Control - Dial-In access	NA	✓	NA
Remote Node Software Requirements	TCP/IP Software with SLIP or PPP	NetWare and Telebit supplied PPP driver	ARA Software available from Apple Requires System 7
Modem Pooling	✓	✓ Requires ACS option	
Security, Management and Configuration			
Security - ID/Password	✓	✓	✓
Security - Callback	✓	Not supported in Release 2.0	✓
Security - Cryptographic Handshake	Only between two NetBlazers	Also between NetBlazer and remote PPP user	Only between two NetBlazers
Security - Filtering	✓	SAP Filtering - Static	ARA Zone Filtering
Configuration - Command Line	✓	✓	✓
Network Booting (Bootp)	✓		
Accounting	✓	✓	✓
SNMP Management - MIB II, Enterprise MIB, and Modem MIB	✓	✓	✓

IPX Support

The NetBlazer has the ability to route IPX packets over any of its interfaces, dial-up, leased line, or local (Ethernet or Token Ring). Ethernet support includes Ethernet II frames or Novell's 802.3 frames while Token Ring support is for Token Ring or SNAP frames. Additionally, Telebit provides a remote PPP shell so that users can run NetWare on a remote computer just as if it were running on a local computer. Users on one LAN can also have transparent dynamic access to file servers on any other LAN. The Telebit White Paper "Dial-Up Routing in a NetWare Environment" has more details on the NetBlazer's IPX support.

Features	Benefits
IPX routing.	Transparent LAN-to-LAN dial-up connections allow NetWare users to get access to remote LAN resources such as printers or file servers.
Compatibility with Novell's IPXWAN specification.	Ensures compatibility with an upcoming standard in the IETF and hence more choices and investment protection for the customer.
PPP Driver - The PPP driver is bundled with the NetBlazer for free.	Standalone remote PCs can run NetWare as if they were connected to a local LAN.
The IPX header packet is compressed to 2 byte from its original 30 byte size.	Reduces the size of the IPX header so that more dial-up bandwidth is available for user data.
Keep-alive (watchdog) packet spoofing - Prevents keep-alive packets from bringing up a dial-up link or keeping a dial-up link open.	This keeps communications costs down by not bringing the link up for keep-alive packets.
Static SAP tables - SAP packets are not sent over dial-up links, but tables can be set up so that a remote NetBlazer can broadcast the services of devices on a local LAN. This allows remote devices to know about and use local services such as printers or file servers.	Keeping SAP packets on the local network also saves on communications costs and increase utilization of available bandwidth.
IPX Tunneling - Users with a TCP/IP backbone can send IPX traffic encapsulated in TCP/IP with the NetBlazer doing the encapsulation and unencapsulation.	Permits customers to use NetWare 2.x and 3.x networks to send IPX traffic on a TCP/IP backbone.This is especially useful when customers have a NetWare 2.x networks on one side that don't do encapsulation and NetWare 3.x on the other side which does encapsulation.
NetBlazer/ACS Option - This option allows remote control dial-in and dial-out modem pooling for NetWare networks.	Customers can get routing, dial-in remote node, dial-in remote control and dial-out all with the same product providing the most flexible and complete NetWare solution.
Routing based on tick counts rather than hop counts as implemented in Novell routers.	The NetBlazer is compatible with Novell routers and can save money by sending data over fixed lines where they exist.

AppleTalk and ARA Support

The NetBlazer has the ability to route AppleTalk Phase 2 packets over dial-up, leased line, or Ethernet. The NetBlazer supports Apple's routing protocols RTMP, ZIP and NBP. The NetBlazer does not route packets over Token Ring or LocalTalk interfaces. If LocalTalk support is desired, a product like FastPath that provides LocalTalk to Ethernet conversion will be required. The NetBlazer does not support AppleTalk Phase I which has been superseded by Phase II. The addressing in AppleTalk is not as complex as IP. The NetBlazer can be configured as a Seed router as well as a Nonseed router. Nonseed routers learn their routes automatically from the Seed routers.

The NetBlazer has ARA protocol support which allows remote Macintosh computers to dial in and use the resources available on AppleTalk networks. ARA protocol permits a full network connection for the remote users. ARA protocol software is available from Apple Computer. In order to use ARA with the NetBlazer, Telebit's T3000 or WorldBlazer modems are required at the NetBlazer end (running LA 7.03 or later revision of firmware on these modems). The remote site Macintosh user can use any modem supported by ARA.

The users on an AppleTalk network can use NetBlazer as a modem pooling device to dial out to bulletin board services provided their communication software can run Telnet program over AppleTalk. Some of the terminal emulation software such as PacerTerm provide this capability. The NetBlazer can also be configured from a Macintosh using PacerTerm software.

Features	Benefits
Any port can be used as an ARA (AppleTalk Remote Access) server as well as AppleTalk router.	One product can provide both functions saving money and easing management.
Multiport ARA (AppleTalk Remote Access) server.	Flexible, Scalable and most cost effective central site server for dial-up access to an AppleTalk network.
Zone filtering for dial-in ARA users. The administrator can define which zones dial-in ARA users can access.	This security feature can be used to prevent dial-in users from looking at unauthorized file servers.
AppleTalk Phase 2 routing.	Transparent LAN to LAN connections over dial-up give users access to LAN resources such as printers and file servers on remote networks.

Enhanced SNMP Management of Modems and NetBlazer

The NetBlazer release 2.0 supports an Enterprise MIB for the NetBlazer in addition to its existing support for MIB II. The NetBlazer now provides a modem MIB as well.

The modem MIB allows a modem connected to the NetBlazer to be managed by an SNMP network management station. The modem MIB allows a network administrator to remotely monitor a modem using SNMP rather than talking directly to the NetBlazer and modem.

Currently, the modem MIB only manages Telebit modems. However, it is Telebit's intention to be establish its MIB as a standard so that all vendors modems can be managed. In order to establish a modem MIB as a standard, Telebit is working through IETF (Internet Engineering Task Force).

Support for New Asynchronous and Synchronous Cards

The NetBlazer 2.0 software also supports a new asynchronous card in addition to support for the existing NM8 asynchronous card. The new asynchronous card's part number is ASYN8. ASYN8 includes the card as well as a modular terminal adapter (MTA) providing eight external RS-232 ports per module. The external 8-port modules can be connected together so that one slot on the NetBlazer can support up to 24 asynchronous ports. All of the ports can run at 115.2 Kbps.

A new synchronous card is available that has two ports on one card. Multiple cards can be put in a NetBlazer. One of the synchronous ports is capable of supporting T1/E1 speeds while the other can support 64K bps. Customers can run V.25bis on both ports so that the two B channels on an ISDN terminal adapter can be connected to one card. The electrical interfaces supported are V.35, RS-232 and RS-449/RS-530. The default interface on the synch cards will be RS232 unless specified otherwise.

Features	Benefits
Asynchronous card	
Modular Terminal Adapters can be connected together externally	Only one slot is needed to support up to 24 RS232 ports.
Port speeds of 115 Kbps	Investment protection as the interface is suitable for next generation faster modems.
Synch Card	
Two Ports per card.	With one Synch card, one interface can support T1/E1 while the other can be used for ISDN or another 56/64 Kbps leased line.
Supports up to T1/E1 interface speeds.	NetBlazer can be used as a low-cost leased line router
Supports up to 10 synchronous serial ports in increments of 2-ports each rated at 128Kbps.	NetBlazer capacity can be increased as user needs grow. Each port can be used for 2B channels.
Combine Dial-up analog and digital services in the same product.	Investment protection and flexibility to support sites with modems and TAs simultaneously.
Supports TAs and CSU/DSUs with V.25 bis dialing.	V.25 bis allows automatic dialing easing the management.

Enhanced Multi-Line (Inverse Multiplexing)

The new algorithm for the multi-line feature in the NetBlazer is more efficient, making file transfers faster and allows dial-up lines to augment the bandwidth of a leased line. The multi-line feature is specific to TCP/IP and is proprietary to Telebit.

When the NetBlazer is configured for multi-line, it can bring up multiple lines simultaneously in response to packet traffic. Therefore, file transfers that need a lot of dial-up bandwidth quickly will not have to wait while each line is brought up individually.

Dynamic Address Assignment

Asynchronous PPP has provisions to allow a server to negotiate an IP address with a remote host or router. This capability means the NetBlazer can be configured with a pool of IP addresses that is equal to the number of modems. These addresses can then be shared by all of the remote users. This feature is especially helpful in environments where very large number of users need to call into a central location or in cases where users are traveling and need to call into different servers. Note, however, that this does not eliminate the need for defining an interface for every user.

Network Booting

The NetBlazer has the ability to boot over the network the same way that diskless workstations can boot from a file server on a network using Bootp. The primary advantage of this feature when compared with the method of booting from the local floppy disk is that only one copy of the NetBlazer software and the configuration files needs to be kept in one central location. All NetBlazers can obtain all of their relevant files from this one central server. Network booting works only on TCP/IP networks where a Bootp server is available.

ACS Option

ACS (Asynchronous Communications Server) functionality is available as an option to the NetBlazer. The ACS provides dial-out modem sharing and dial-in remote control for Novell IPX LANs. Contact your Telebit sales representative for more information.

Release 2.0 Minor Software Features

Features	Benefits
User password files can be stored on a remote server.	Password files that exceed the capacity of the NetBlazer disk or memory can be used.
TALK enhancements - It is possible to TALK to a modem pool and have the NetBlazer find the first available port.	This allows the serial ports to be connected to the serial ports on a computer so network users can have access to the computer through its serial ports.
DIAL enhancements - Authorized users can issue the dial command followed by any phone number. The NetBlazer will find the first available modem and dial that phone number.	User can dial any phone number without having the system administrator enter the number first.
Line editor.	NetBlazer configuration files stored on the local floppy disk can be edited directly on the disk allowing faster modifications.
TCL Support- A command processor that supports advanced scripts by providing variables and conditional commands. Users must be familiar with the UNIX TCL facility.	More complex functions can be executed automatically.
Packet-mode callback security for IP.	Packet-mode users can have an additional level of security.
Multiple NetBlazer commands per line.	Faster more flexible administration.
User-based command execution - A unique NetBlazer command can be executed at login time on a user basis.	This allows the system administrator to set up custom login scripts so that users can be directed to a specific host via Telnet or rlogin every time they call in.
Modem result codes can be set in the modemcap file.	Increased ability to connect different types of devices such as non-Telebit modems, X.25 PADs and terminal adapters to the NetBlazer.

NetBlazer 2.1 Enhancements

PN Support

NetBlazer 2.1 includes all necessary drivers for the latest NetBlazer hardware platform, the NetBlazer PN (PN1 and PN2). The PN1 has an internal modem which has documentation included as comments (in the modemcap.txt file) that precedes the pn1-m entry.

Kerberos Support on NetBlazer

NetBlazer 2.1 now supports Kerberos version 4, which provides an additional level of security for NetBlazer customers who have a Kerberos server.

Kerberos provides a way to authenticate users via an encrypted exchange over the network. Passwords are never sent as clear text over the network. Also, Kerberos can authenticate the user to the destination host in a telnet or rlogin session. This avoids having the password sent as clear text over the network in response to the password prompt, thus providing greater password security. Users can also update his or her password with the Kerberos server, thereby relieving the system administrator of the task of keeping NetBlazer passwords current.

Kerberos System Requirements

A Kerberos server is required at the LAN. The NetBlazer acts as a Kerberos client and passes the request from the remote user to the Kerberos server on the LAN. The Kerberos server software is distributed by MIT Project Athena at no charge, as a public domain software. Some other companies like Cygnus Support (smlieu@cygnus.com) help users in setting up the Kerberos servers and provide documentation.

Note: *At this time, Kerberos only works on TCP/IP protocols.*

ODI Support in PPP Shell

The NetBlazer 2.1 software now provides an ODI-compliant PPP shell (in addition to a monolithic PPP shell) for the remote IPX users. This facility enables a remote user (with an appropriate ODI-compliant TCP/IP software package) to be logged simultaneously to a NetWare server using IPX and to a UNIX host using IP.

ODI is a specification defined by Novell at the MAC (media access control) layer. The objective of ODI is to enable NIC (network interface cards) vendors to write one set of drivers for their cards that will work with any number and type of transport protocols. At the same time, ODI allows software vendors to develop a transport protocol stack like TCP/IP that will work with any hardware.

Many applications require users to be connected to an IPX server and talk to the database server, which may run on top of UNIX with TCP/IP protocols. In the past, users had to set up complicated batch files or reboot every time they wanted to disconnect from the IPX file server and reconnect to the Unix host or vice versa.

Remote Software Requirements

To enable the remote user to connect to NetWare file servers as well as IP hosts, the user needs the following:

- ODI version of the PPP shell - Supplied by Telebit with NetBlazer 2.1 software.
- TCP/IP software that is compliant with ODI shell.

At the time of release, Telebit has tested the following TCP/IP packages:

- Novell's LAN Workplace for DOS version 4.1

Generally, Telebit suggests that you don't use the ODI-based PPP shell if you are not going to run multiple protocols because the ODI shell occupies about 25KB more DOS memory than the monolithic drivers.

NTP

The NetBlazer 2.1 software now provides a network time service. This facility enables a user to update the NetBlazer's time of day clock by requesting the time from the LAN NTP server.

To use NTP set the gmt offset first: gmtoffset [<hhmm>]. Then use ntp to set the NetBlazer's time:

```
ntp [-s] [-f] [-v] <host>
```

2.1 New Commands and Features

General

- New "where" command now displays the location in the command tree. This is mostly intended for use from aliases and scripts.
- New "bin exit" command was added so that scripts could be aborted more cleanly.
- The gmtoffset command has been moved to configure so that it can be used without kerberos being present. Used with ntp command in setting the NetBlazer's time from a server.
- The ability to trace ethernet interfaces to file in format compatible with "tcpdump -r". E.g. trace en0 dump ofile=tcpdump.en0.
- The msglog command has been improved (can now set level and can accept multiple words).

IPX and AppleTalk Commands

- A new parameter has been added to the **appletalk route learn** command that specifies the number of seconds the NetBlazer waits for RTMP and ZIP broadcasts to be received before learning the routes to the far side. This time should not be set lower than its default value of 120 seconds unless the link is already up.
- The new **ipx route learn** command allows you to get a snapshot of the IPX routes on a remote dial-up network. The **ipx route forget** command removes IPX routes acquired through the **ipx route learn** command.

Debug Commands

- A "debug chat" was added to display the names of the available chat scripts and inquire as to their contents.

- The debug "files" command has been renamed to debug "openfiles" to avoid conflict with the configuration "files" command.
- Debug command arguments can now be simple expressions, e.g.

  debug db 0x1235+5
- The "debug status" display now shows "uptime" and handles displays longer than the screen size better. Also allows fractional delay values.
- Crashes are logged via syslog to the host that network dumps will be taken to, or via a broadcast, or to the address specified via debug crashlog on | off | host <host> | ether <ether> | level <level> this makes it easier to diagnose crashes at sites that do not have network core dumps set up. The level defaults to critical.
- debug dump now takes a core dump and then the NetBlazer continues running.
- debug remote and debug crashlog are better about figuring out which ether address to use (e.g. if the route is through a router they do an arp lookup for the router). This makes it less necessary to specify the ether parameter.
- debug remote (and remote test) display the return code from the tftp server to make it easier to figure why the host has rejected the request.
- debug remote (and crashlog) will attempt to verify that the host specified is on the local network (e.g. consistent with the ethernet specified) and will set the ether address to that of the appropriate gateway if the "ether" command isn't used. This should make it easier to configure remote dumps.
- The watchdog timer on an Ethernet port is enabled by

  debug watchdog <en0|en1|en2> <delay>

  will reset the specified ethernet if no input packets are received within <delay> seconds and output has been sent recently. This should only be necessary for UTP connections that have been seen to hang.

  To disable the watchdog: debug watchdog <en0|en1|en2> 0

  the command: interface list <en0|en1|en2> now shows the idle time.

Files Command

- config file end used to execute commands after the end of startup.cnf while the system is coming up multi-user (e.g. "end.cnf").
- The command "config file <file>" now just shows the name of the file, without the <file> prefix

- config file <file> {command} allows a command to be specified instead of just the name of a file containing commands. This can be particularly useful for end, e.g.

  config file end "tcl {while (1) {ntp -s ntp; sleep 3600}}"

  where the ntp host is "ntp". This uses ntp every hour to bring the time into sync with the ntp server.

Syslog Commands

- "syslog host on <hostname> <facility>" no longer requires a "syslog host off" if there already is a syslog host in effect.
- If an invalid "facility" is given a list of valid ones is listed.
- The output from the list command has been improved.
- Syslog now provides dialout usage time reporting. Previously there was no record logged to syslog indicating how long a user was accessing a dialout modem. This is information is now reported for accounting purposes.

Modemcap

- new modemcap entry "line-length" that allows a maximum length specification for the cmd-list output for particularly verbose modems.
- The modemcap.txt notes on autoconfiguration (near the top of modemcap.txt) now describe special cases for modem ATI responses.
- The modemcap file now contains more modems and TA's, including Octocom V.32 and V.32 bis modems.
- Support for PN1 and PN2 is now included. The "PN1" is a NetBlazer with an internal modem. Documentation for the PN1 is included as comments that precede the pn1-m entry.
- Default time-outs in the state machine can be overridden in the modemcap.txt (this is particularly relevant for Switched-56 DSU/CSU's).
- The QBlazer+ modem is now supported.
- WorldBlazer Rack modemcap entry now works [2.0 had a bad speed setting] and has Q command in reset string.

Others

- The login messages now show up even when the login is started while the system is still configuring.
- The "line list" output shows the modem name in full, while the old list could truncate it.

- Aliases can now be longer and are saved properly rather than being truncated at 131 characters.
- Dumping the state of the NetBlazer and restoring service does not take as long. Users are not forced off.
- Crypto keys have been suppressed from the user list output.
- The number of login attempts with a bad user id is now limited
- Modify the behavior of the commands:

  configure ip address
  configure ip netmask
  configure ip broadcast

  such that an arp entry is automatically added for ether and token interfaces corresponding to the current broadcast address. Commands of the form:

  "ip address xx.xx.xx.xx/xx en0"

  will both add the appropriate arp entry for broadcast, as well as add a route for the interface. The arp entries added for all of these commands will be deleted from the arp table if the broadcast addresses of the associated token or ether interfaces are changed. Also, the arp entry added for broadcast address will correctly reflect the type of interface (previously, token interfaces were treated as ether).
- IP RIP Changes: The netblazer does a much better job of keeping the IP routing table organized properly. Things that used to be problems that are now fixed include:

  - The NetBlazer wouldn't always use the least cost route to a destination
  - Sometimes the wrong route would get clobbered by RIP updates.
  - RIP routes sometimes get deleted completely rather than going into hold-down when they time out.
  - If the NetBlazer clock is changed, RIP routes could take a long time to time-out.

IPX Configuration Now Semi-automated

The NetBlazer now discovers the IPX network number, IPX framing, and assigns a unique internal IPX network number. When no startup.cnf is found, the default.cnf file causes the NetBlazer to enter IPX startup mode. The NetBlazer continues in IPX startup mode until a save is done.

During IPX startup mode, all LAN interfaces are enabled for IPX. When a packet is observed on an interface, it's network number is used to set the network number of the interface.

During IPX startup, their initial framing is 802.3 for Ethernet and token-ring Token-Ring. If no packets are received with this framing for one minute, we try the other framing ethernet_2 for Ethernet and token_ring_snap for Token-Ring. It continues to toggle the framing on these interface until packets are received or a save is done.

During IPX startup, it will try to assign a unique internal network number. This number is 4 bytes long, where the first byte is 0xFF. The next 3 bytes are the last three bytes of the mac address of a lan adapter. It uses the mac address of the last lan adapter found in this sequence: en0-2, tr0-2.

Additional Documentation

Please note the following changes and additions to your NetBlazer 2.0 documentation set.

dtrtime and dsrtime settings

Telebit recommends that dtrtime be set to 3000 milliseconds and that dsrtime be set to 1.000 second for use with NetBlazer 2.0 software. Use the NetBlazer command SOURCE UPDATE.CNF to automatically update dtrtime and dsrtime for all of your asynchronous lines. In no case should dtrtime be set less than 100 milliseconds.

Dedicated interface watchdog

Note that the watchdog that supports backup for dedicated interfaces is enabled in NetBlazer 2.0 using a **server start** command. While the default interval (120 seconds) for verification of operation of the interface is appropriate for most environments, applications that desire faster switchover to a backup interface may set a shorter time. Few applications will require a shorter time than 20-30 seconds. Note that shorter intervals increase overhead traffic on the interface.

Line terminal command

Note that the NetBlazer does not directly use the <terminal-type> parameter. This information is passed to a telnet/rlogin server by the NetBlazer's telnet/rlogin clients, and allows the UNIX host to set the "TERM" environment variable. Hence, the value used in this parameter should be one that is installed in the host's /etc/termcap file or /etc/terminfo database.

Avoiding call collisions on dynamic interfaces

Whenever two NetBlazers are configured such that either can call the other, there exists a danger that the two calls will occur at the same time, collide, and both calls will fail. There are two ways to avoid this situation. First, if the application allows, configure one NetBlazer to receive the calls and the second to place the calls. This eliminates the possibility of "glare" conditions.

If the application requires both NetBlazers to be able to place a call to the other, Telebit recommends that the minwait time be set to 5 seconds and the maxwait time be set to 100 seconds. This will provide a wide range for the random backoff timer, and thereby increasing the probability of success.

The tips above are especially important for IP multiline operation, where multiple calls are aggregated to provide a single logical link between two NetBlazers.

Learning IPX routes and SAP information via a dynamic link

Telebit recommends use of static IPX routes and SAP entries for dynamic links. If you have a very large number of IPX routes and SAP entries, you can utilize the following procedure to establish this information automatically. The drawback to this approach is that it is not possible to save the learned information in the STARTUP.CNF file.

First, add an IPX route in each NetBlazer to the other NetBlazer's internal network number through the dynamic interface. Enable RIP and SAP broadcasting on that dynamic interface. Use the **ipx ping** command, specifying the other NetBlazer's internal network number and a node number of 1, to establish the link. When the link goes down due to inactivity, IPX routes and SAP information learned from the other side will not be aged.

NetBlazer memory recommendations

The NetBlazer needs a certain amount of free RAM to allocate buffers and create temporary processes. NetBlazers with 2 megabytes of RAM that routinely have less that 250 kilobytes free will perform better if upgraded to 4 megabytes.

The NetBlazer administrator can check how much RAM is free by using the **mem** command. The first three lines of output from the **mem** command look like this:

```
Memory usage status:
text start 8000, data start a1000, bss start b1b80, end c4e68
heap base c4e68 size 1282456 avail 675108 (52%) low mark 655704 (51%)
```

The third line is the one to pay attention to:

avail <available bytes> (per cent) low mark <lowest available bytes> (percent)

If avail frequently falls below 250,000 bytes or the low mark falls below 100,000 bytes, then the NetBlazer may need a memory upgrade, or you may unload any unused functions (see Recovering memory by unloading SNMP and Recovering memory by unloading AppleTalk on the following pages).

If you need to add memory to your NetBlazer, contact your Telebit sales representative or contact Telebit Technical Support at 1-800-TELEBIT or 408-734-5200.

AppleTalk ARA requirements

It is necessary to set dcdtime for each line to be used for AppleTalk ARA support to at least 4.0 seconds. Telebit modems are required for the NetBlazer side of the ARA link, and must use LA7.03 or later firmware.

Use of same user ID by multiple dial-in users (ARA, SLIP, and PPP interfaces)

It is important to note that the NetBlazer does not support the use of a single user ID by multiple concurrent dial-in users for dynamic interfaces, including SLIP, PPP, and ARA connections. Note that, in the case of ARA, the error message (bad user) may be misleading in the event that two users attempt to use the same user ID.

Configuration of PPP interfaces for AppleTalk support

It is important to note that the use of PPP interfaces for AppleTalk routing is set entirely by the appletalk interface commands **add, delete, enable**, and **disable**. ATCP negotiation for PPP links is appropriately setup by the NetBlazer automatically. It is not necessary to use the **ppp protocol** command to control ATCP use.

Resetting the homezone for an ARA user

If you wish to reset the homezone for an ARA user to the initial state (where no homezone has been established), use the command **configure user homezone** <user> "", where "" represents the null string.

Recovering memory by unloading SNMP

NetBlazer 2.0 loads SNMP support by default. If your configuration does not require the use of SNMP, you can uninstall SNMP in order to free system memory for other uses. Use the following commands to unload SNMP, and then save and reboot the NetBlazer.

(for some 1.5 upgrades)

```
snmp SysContact
snmp SysLocation
snmp SysName
snmp community delete SNMP_trap
```

(for new 2.0 installation and most 1.5 upgrades)

```
snmp community delete public
snmp community delete private
server stop snmp
```

(for new 2.0 installations)

```
server stop ipxsnmp
snmp enterprise disabled
```

Save and reboot.

Recovering memory by unloading AppleTalk

NetBlazer 2.0 loads AppleTalk support by default. If your configuration does not require the use of AppleTalk, you can uninstall AppleTalk in order to free system memory for other uses. Use the following command to unload AppleTalk, and then save and reboot the NetBlazer.

```
configure appletalk interface delete en0
```

Save and reboot.

Printing to an AppleTalk laser printer

When a laser printer on an AppleTalk network is powered up before any router is available on the network, it may not be possible to print to that network printer (even though it appears in the Chooser). This is an addressing problem, and will occur with any AppleTalk router.

Reset or power cycle the printer after starting a router to resolve the problem.

Extraneous "en0: output busy timeout" messages

The message en0: output busy timeout indicates that the NetBlazer Ethernet controller had stopped transmitting data and was automatically restarted. No operator intervention is required, and this message can be safely ignored.

Recovering modems that the NetBlazer has flagged as "broken"

When a line's USE field is set to broken by the NetBlazer, this indicates that the NetBlazer's attempt to identify and configure the modem using the MODEMCAP.TXT file failed. The likely causes of this are a defective modem or a data rate mismatch between the modem and the line. To determine if the problem is data rate related, manually set the modem type and set the line to autoconfigure (no auto-identify) using the **line auto linexx config** command. This causes the NetBlazer to try all reasonable data rates, and should resolve any data rate mismatch problems.

Difficulty transferring binary files using ZMODEM with TNGLASS

If you experience problems when transferring binary files using ZMODEM with TNGLASS, try using binary telnet.

Version 2.1 Fixed Bugs

IPX Log command options "ipx", and "spx" removed

The IPX Log command previously showed six facilities for logging: acs, ipx, rip, net, spx, and sap. All six options were displayed when log was typed, but only acs, rip, and sap were actually implemented. The "ipx" and "spx" option have been removed from the "log" command and the usage has been modified. Note that "net" is still in there, since it is used to debug IPX dial and PPP interfaces.

The number of login attempts with a bad User ID is now limited

The NetBlazer now disconnects a user that gets the user ID wrong too many times. If a valid user ID and an invalid password is entered, three attempts are allowed before disconnect occurs. The invalid user ID attempt is counted towards the limit. For example, two bad User ID attempts followed by one bad password attempt results in a disconnect.

Dialback destination number now concealed

The NetBlazer no longer announces the destination phone number in call-back after the "Where:" prompt.

Modem pooling sessions no longer gobble the escape character

Modem pooling sessions used to "absorb" the escape character that should have been passed to the remote host. This problem is now fixed.

v.25bis time-outs now configurable to work with switched 56 service.

Previously, v25.bis dialing did not work with switched 56 services because the time-outs were too short.

Precautions 3

Version 2.0 Precautions and Known Problems

Telebit strives to provide a bug-free product to its customers. There are, however, several restrictions and limitations in this software release. A number of the important precautions are described in this section. These issues are under examination at this time for determination of changes for a future NetBlazer release.

Interface backup does not detect modem or DSU in loopback mode

If a modem or DSU is placed in loopback mode on an interface for which the automatic backup capabilities of the NetBlazer are configured, the NetBlazer will not activate the alternate line. All other failure modes for the modem or DSU are properly handled.

Modem setup command sent to line with autoconfigure set to off

In certain (very rare) circumstances, it is possible that the NetBlazer may send an AT command to a line for which autoconfigure is set to off. This is ordinarily not a problem. Inserting a 1-2 second delay in the login sequence for the host that is connected to the line should completely eliminate any remaining symptoms.

Memory leaks resulting from killed TCL scripts

In some circumstances, killing an executing TCL script may result in unreleased memory. In extreme circumstances, if many TCL scripts are killed, it may be necessary to restart the NetBlazer to recover lost memory. TCL scripts that are allowed to run to completion do not result in any memory leaks.

NetBIOS applications and the IPX PPP shell

A NetBlazer bug is currently preventing NetBIOS applications from operating correctly over the IPX PPP shell. If your environment requires such applications, please contact Telebit Technical Support.

LIPX NLM not recommended for sync interfaces between IPX token ring LANs

The LIPX (large IPX) NLM for Novell file servers is not recommended for NetBlazer synchronous links used to connect IPX token ring LANs.

AppleTalk internetwork size limitation

The NetBlazer has been tested to approximately 100 networks when performing AppleTalk routing. If the NetBlazer is to be installed in an internetwork with more than 100 AppleTalk networks, use of the nonroute configuration is recommended.

AppleTalk traffic not automatically restarted after sync link recovered

When a synchronous link fails and then subsequently restarts, AppleTalk traffic may not automatically restart on the link. To restart traffic flow over the link, it may be necessary to use the commands below:

interface appletalk disable <sync-interface>
interface appletalk enable <sync-interface>

Version 2.1 Precautions and Known Problems

When using 2.1 of the NetBlazer software, you should be aware of the following known limitations or restrictions.

WorldBlazer Reset and Configuration Workarounds

WorldBlazers with LA7.02 firmware or above take significantly longer to reset after DTR goes down. These modems do not work well in answer mode with version 1.5. Steady progress has been made in addressing this problem in version 2.0 and version 2.1. The following procedure is recommended for avoiding reset and configuration problems with WorldBlazers:

1. Reset the modem, by powering off, holding in T/D, then powering on, etc.
2. Set the line speed on the NetBlazer to 9600 bps.

   "speed line10 9600"
3. Set autoconfigure to "config" for each line:

   "auto line10 config"
4. Set the modem type to "wblazer" for standalone WorldBlazers or "wblazer-rm" for rack-mount WorldBlazers, for each line:

   "modem line10 wblazer"
5. Make sure DTR time is greater than or equal to 3000 ms for each line, since it takes about 3 seconds for a WorldBlazer to reset (if the WorldBlazer is running 7.02 code or later).

   "dtrtime line10 3000"
6. Tell the NetBlazer to ignore DSR changes:

   "dsrtime line10 0"

7. Reset each line:

"reset line10"

IP RIP can only be sent to a reachable address

You can't have a RIP distribution host entry unless the NB thinks it's reachable.

System Update Procedure 4

Using Diskcopy to Update Your System

To create a new working copy of the NetBlazer system diskette with your current system configuration using the **diskcopy** command, perform the following steps.

1. Start the NetBlazer using your current working copy of the NetBlazer system diskette, and log in as "root" or as some other user with system configuration privileges.
2. Format a new high-density diskette, and make a backup disk copy of the new system distribution diskette. Store the original NetBlazer system diskette in a safe place.
3. Save your current system configuration to your backup copy of the system diskette by entering a **configure save** command at the command prompt.
4. If you have customized your CHAT.TXT or MODEMCAP.TXT files, you will need to apply your modifications to the new files using a text editor on an MS-DOS system, or use FTP to transfer the files to a system with a text editor, modify the files and then transfer the files back to your NetBlazer using FTP.
5. Reboot your NetBlazer by entering a **reboot** command at the command prompt, and log in again as "root" or as some other user with system configuration privileges.
6. If you are updating from a version earlier than NetBlazer 2.0, enter **source update.cnf** at the command prompt to update your current configuration.
7. Save your current system configuration to the system diskette by entering a **configure save** command at the command prompt.

Command Reference Manual Updates 5

Replace or add the following sections in your *NetBlazer Command Reference Manual*:

- BIN - Bin Commands
- CON - Configure Command
- DEB - Debug Commands
- FIL - Files Command
- GMT - Gmtoffset Command (new)
- INT - Interface Commands
- IPX - IPX Commands
- KER - Kerberos Commands (new)
- MSG - Msglog Command
- SES - Session Commands

Bin Commands

The **bin** command takes you to the Bin command level, which comprises a collection of miscellaneous commands. To view a list of the commands available at this level, type a question mark (?) at the Bin command level.

```
NB:Top> bin
NB:Top>Bin> ?
Available commands:
commands        echo            edit            eval
exit            grep            kdestroy        kinit
klist           kpasswd         more            output
passwd          settime         sleep           source
tee             tcl             time            top>
tty             type            up>             uptime
version         wall            where           who
write           ?
```

bin commands [terse|verbose]

The **commands** command displays a complete list of the available NetBlazer commands. If you enter the optional terse parameter, all commands at a particular command level are displayed on a single line rather than on multiple lines. This difference is illustrated in the following example.

```
NB:Top>Bin> commands |grep Alias
Top>Configure>Alias>  add
Top>Configure>Alias>  delete
Top>Configure>Alias>  list
Top>Configure>Alias>  top
Top>Configure>Alias>  up
Top>Configure>Alias>  ?
NB:Top>Bin> commands terse |grep Alias
Top>Configure>Alias>  add delete list top up ?
```

bin echo [-n] <message>

The **echo** command prints a message to the user's screen. If you do not include the optional -n parameter, the message is followed by a new line. However, if you include the -n parameter, the new line is not sent. A couple of examples are shown below.

```
NB:Top> echo Hello NetBlazer users
Hello NetBlazer users
NB:Top> echo -n Hello NetBlazer users
Hello NetBlazer usersNB:Top>
```

Echo commands are useful in source files to give users feedback about what is happening when they run the **source** <filename> command.

Another use for the **echo** command is to set up a login message for a specific user by using **echo** in the user's login command, as illustrated below.

```
NB:Top> user command carol "echo Access denied - Call MIS department; logout"
```

bin edit [<filename>]

The **edit** command allows you to edit a local or remote file. Using the NetBlazer editor requires an understanding of the UNIX ed and ex line editor commands. Refer to the UNIX man pages for a description of these two commands.

Caution: *The **edit** command should only be used to edit ASCII files that contain 1000 lines or less.*

Do not attempt to edit TELEBIT.SYS, FRED, or any of the loadable module (.LDR or TOKEN002.MCD) files.*

Before editing any of the NetBlazer configuration files (STARTUP.CNF, MODEMCAP.TXT, CHAT.TXT or PASSWD), be sure to make a backup copy of the file.

To start editing a file, enter **edit** followed by the name of the file at the NetBlazer command prompt.

```
NB:Top>Bin> edit startup.cnf
"startup.cnf" 211 lines, 7851 characters
*
```

To edit a file on an remote IP host system, use **edit** <host>:<pathname> to specify the location and pathname of the file.

When the editor is started, it loads a copy of the file into a special area in the NetBlazer's memory called the buffer. Any changes made in the buffer do not appear in the original file until you enter a command to write (**w**) the contents of the buffer to disk.

To close a file without saving any changes, enter **q!** at the editor command prompt.

```
*q!
NB:Top>Bin>
```

The NetBlazer editor provides various line oriented editing commands that are used to display, change, delete or insert lines in the file's buffer. To view a summary of the editor commands, enter **h** at the editor prompt.

```
*h
(.)a                append after given line
(.,.)c              change lines
(.,.)d              delete lines
e[!] file                   edit new file
f file              file information or new file name
(1,$)g/pattern/cmd          execute cmd for all lines that match
h                   help information
(.)i                insert text
(.,.+1)j            join lines
(.)kx               create mark
(.)l                list lines
(.)p                print lines
q(!)                quit editting file
($)r [file]         read from file
(.,.)s /s1/s2/g     substitute
set [no]number [no]ignorecase [no]prompt all
(1,$)v/pattern/cmd          execute cmd for all lines that don't match
version             display version number
(1,$)w[!] [file]    write to file
(1,$)wq [file]      write to file and quit

line numbers (lnr)'s:
number              decimal line number
/pattern/           find pattern string
?pattern?           backwards find pattern string
$                   last line
.                   current line
lnr[+-][number]     above + or - decimal number or 1
```

Users of the UNIX ed or ex line editors should recognize most of the commands listed above. Note, however, that not all of the features of ed or ex are implemented in the NetBlazer editor. For example, the use of regular expressions for pattern matching is not supported in the NetBlazer.

When using a line-oriented editor, you must specify which line or lines to operate on and the operation to be performed. If only one line number is specified, the operation is performed on that line. If two line numbers separated by a comma are specified, the operation is performed on the inclusive range of lines. For example, **1,5p** means print (display) the first five lines.

The **$** character is used to address the last line of a file. Thus, entering **1,$p** prints all the lines in a file.

The editor keeps track of the line on which you last performed some operation. This line is called the current line and is addressed by typing a dot character (a single period).

If you do not specify a line number, most edit operations are performed on the current line. For example, entering a **d** command without specifying a line number deletes the current line (the last line addressed).

If you do not specify an operation, the editor prints the line or lines specified by the line address. Therefore, entering **1,$** is equivalent to entering **1,$p**. If you press the return key without entering a line address or operation, the editor selects and displays the line following the current line.

You can specify a line number that is relative to the current line by using the + (plus) and - (minus) characters. For example, **+5** refers to the line five lines after the current line; **-3** refers to the line three lines before the current line. Entering a + or - without a specifying a number refers to the next or previous line.

One of the most convenient features of the editor is the ability to reference a line based on its content. The following display illustrates how to use this feature to find lines in the STARTUP.CNF file that contain the text pattern "route."

```
NB:Top>Configure> edit startup.cnf
"startup.cnf" 211 lines, 7851 characters
*/route/
#add static routes:
*//
configure ip route addprivate 127.0.0.1/32    loopback
*//
configure ip route addprivate 143.191.20.1/32    tbupdate
*//
configure ip route add 143.191.11.0/24    en0
*//
#add static routes:
```

The editor starts at the line following the current line and searches forward in the buffer looking for the first occurrence of the text pattern "route." If the end of the buffer is reached, the editor wraps around to the beginning, and continues searching until the pattern is found or it ends up back at the current line. If the text pattern is not found, the editor displays the message Pattern not found.

Notice that entering // is a shorthand way to search for the next occurrence of a previously specified text pattern.

You can search backward through the buffer by enclosing the search string between question marks in place of slashes, as shown below.

```
*?route?
configure ip route add 143.191.11.0/24    en0
```

The next several examples shows you how to create and edit a new file. The new filename must conform to DOS file naming conventions – eight alphanumeric characters with a three character extension. Do not use control characters or special symbol such as: *?:;,=+<>\"[]\t

```
NB:Top> edit example.src
Can't open file 'example.src'
*
```

Because this is a new file that does not exist on the disk, the editor complains that it cannot open the file. If you get this message when you attempt to edit an existing file, it indicates that you probably have made a mistake when entering the filename.

The asterisk (*) following the opening message is the editor's command prompt, which tells you that the editor is prepared to accept commands.

When editing a new file, the first thing to do is to start adding text to the file using the append (**a**) command. Whenever you use a command that requires you to enter new text, the editor goes into input mode and waits for you to type the new text.

```
*a
echo This is the first line
echo This is the second line
echo This is the last line
.
*
```

To let the editor know that you are finished entering text, you must type a period at the beginning of a line, followed immediately by a carriage return. This returns you to the command prompt, where you can enter commands again.

Next, let's assume that you want to add a new line before the first line in the file. To do this you have two choices. You can use the **0a** command to add the new line *after* line 0, or you can enter **1i** to insert the new line *before* line 1. The next display shows you how to insert a new line before line 1.

```
*1i
echo This is the new first line
.
*
```

As shown in the following display, you must now change the second and third line in the buffer to read correctly.

```
*1,$
echo This is the new first line
echo This is the first line
echo This is the second line
echo This is the last line
*
```

One way to do this is to use the change (**c**) command to replace the second and third line.

```
*2,3c
echo This is the new second line
echo This is the new third line
.
*
```

Another way to do the same thing is to use the substitute (**s**) command to edit the second and third line, as shown below.

```
*2s/first/new second/
*3s/second/new third/
```

These two commands tell the editor to replace "first" with "new second" in line 2 and replace "second" with "new third" in line 3.

Displaying the buffer contents shows that the file is now correct and ready to be saved to disk.

```
*1,$
echo This is the new first line
echo This is the new second line
echo This is the new third line
echo This is the last line
*
```

The write (**w**) command saves the current contents of the buffer to disk and returns the command prompt to allow you to continue using the editor.

```
*w
"example.src" 4 lines, 116 characters
*
```

If you are finished editing, use the **wq** command to save the file to disk and quit out of the editor.

```
*wq
"example.src" 4 lines, 116 characters
NB:Top>
```

bin eval <command>

The **eval** command is used in **tcl** scripts to evaluate **netblazer** commands that include pipes or aliases. See the **tcl** command description later in this section for a description of **tcl** scripts.

bin exit

The **exit** command is used to exit a source file or **tcl** script.

bin grep <string> [<filename>]

The **grep** command searches a file or the piped output of a command for all occurrences of a specified string.

To display all lines in a file that contain a specified string, use the **grep** <string> <filename> command. For example, the following command displays a list of all the **route** commands in the STARTUP.CNF file.

```
NB:Top> grep route startup.cnf
#add static routes:
configure ip route addprivate 143.191.20.1/32    tbupdate
configure ip route addprivate 127.0.0.1/32    loopback
configure ip route add 143.191.10.0/24    en0
configure ip route addprivate 143.191.22.0/24    en0  143.191.10.29  3
configure ip route add default en0  143.191.10.253  4
```

To scan the output of a NetBlazer command for lines containing a specified string, use the **|grep** <string> command. The next example displays a list of routes that use IP address 143.191.10.29 as a gateway.

```
NB:Top> route list |grep " 143.191.10.29"
143.191.13.0    24  en0          143.191.10.29    2       P 0      0
143.191.22.0    24  en0          143.191.10.29    3       P 0      0
```

When searching for a <string> that includes one or more spaces, you must enclose the string in quotes, as illustrated in the above screen. When searching for a <string> that includes an exclamation point, you must enter it as \! to avoid confusion with a **history** reference.

If the output of **grep** produces more than a screenful of data, you can pipe it to **more** to display one screenful at a time (e.g., **route list |grep 11.2 |more**).

Not all commands can be piped through **grep**; in particular, any command that starts a session (e.g., **ping** or **telnet**) and **trace** commands cannot be piped through **grep**.

bin kdestroy [-f] [-q]

The **kdestroy** command destroys the user's active Kerberos authorization tickets. The -f option causes **kdestroy** to run without displaying a status message. The -q option prevents the terminal from beeping if it fails to destroy the tickets.

bin kinit [-irvl] [<username>]

The **kinit** command allows registered Kerberos users to log in to the Kerberos authentication system. Note that Kerberos users are automatically log in to Kerberos when they log on to the NetBlazer. So, they only need to use this command when their original tickets have expired.

If you enter a **kinit** command without specifying a user name, the NetBlazer prompts you for the user name and password.

The **knit** command includes the following options:

- -i Prompts you for a Kerberos instance.
- -r Prompts you for a Kerberos realm.
- -v Display a status message indicating the success or failure of your login attempt.
- -l Prompts you for a ticket lifetime in minutes.Due to protocol restrictions in Kerberos Version 4, this value must be between 5 and 1275 minutes.

bin klist

The **klist** command prints the identity of the principal, and lists the principal names of all Kerberos tickets currently held by the user, along with the issue and expire time for each authenticator. Principal names are listed in the form name.instance@realm, with the '.' omitted if the instance is null, and the '@' omitted if the realm is null.

bin kpasswd [-n <user>] [-i <instance>] [-r <realm>] [-u <fullname>]

The **kpasswd** command changes a Kerberos principal's password.

If the -n option is specified, <name> is used as the principal name rather than the username of the user running kpasswd.

If the -i option is specified, <instance> is used as the instance rather than a null instance.

If the -r option is specified, <realm> is used as the realm rather than the local realm.

If the -u option is specified, a fully qualified kerberos principal can be given.

The **kpasswd** command prompts for the current Kerberos password (printing the name of the principal for which it intends to change the password), which is verified by the Kerberos server. If the old password is correct, the user is prompted twice for the new password. A message is printed indicating the success or failure of the password changing operation.

bin more [<filename>]

The **more** command displays the content of a file or the piped output of a command one screenful (23 lines) at a time. When the screen is full, you are prompted to press any key except the **q** key to display the next screenful. Pressing the **q** key stops the display and returns the command prompt.

To display the contents of a file, use the **more** <filename> command. For example, entering **more passwd** displays the contents of the PASSWD file one screenful at a time.

To display the output of a NetBlazer command one screenful at a time, use the **|more** command. For example, entering **route list | more** pipes the output of the **route list** command through **more**.

Not all commands can be piped through **more**; in particular, any command that starts a session (e.g., **ping** or **telnet**) and **trace** commands cannot be piped through **more**.

bin output [-a] <filename>

The **output** command sends the piped output of a command to a file that you specify in the command string. You can specify a new or existing file name. If you select a new file name, the **output** command creates that file and stores the piped output there. If you specify an existing file, the new information writes over the existing file unless you precede the file name with the -a parameter. If you include -a in your command string, the **output** command appends the piped output to the end of the file.

The following example shows how to use the **output** command to save the current interface trace settings.

```
NB:Top>Configure>Interface> tracedump |output trace.src
```

Later, you can restore the current interface trace settings by entering a **source trace.src** command.

bin passwd [<username>]

The **passwd** command allows all local users to change their password, provided the system administrator has enabled this feature by issuing a **user changeable on** command.

Note: *This command cannot be used to change the passwords of remote users (i.e. those users whose user records are kept in a passwd file on a remote host). Remote user records may only be changed on the remote system.*

Normal (nonprivileged) users can change their password by logging into the NetBlazer and entering the **passwd** command. They are then prompted for their old password and the new password.

```
NB login: carol
Password:
NB:Top> passwd
Changing password for carol
Enter old password :
New Password :
Retype Password :
```

The optional <username> parameter may only be entered by a user with sysconf privileges to change the password of another user.

```
NB login: root
Password:
NB:Top> passwd carol
Changing password for carol
New Password :
Retype Password :
```

This has the same result as using the **user password** command to change a user's password.

bin settime

The **settime** command sets the date and time used by the NetBlazer.

```
NB:Top>Bin> settime
Current time is Sun Oct 11 19:25:13 1992
[you may enter a non-numeric value (e.g. "cancel") to cancel data entry]
Enter the year [92]:
Enter the month number (1 to 12) [10]:
Enter the day of the month (1 to 31) [11]:
Enter the hour (0 to 23) [19]:
Enter the minute (0 to 59) [25]:
Enter the current second (0 to 59) [13]:
```

Entering a **time** command displays the current date and time.

bin sleep <delay>

The **sleep** command tells the NetBlazer to wait the number of seconds specified by the <delay> parameter before processing the next command. This command is useful in **source** command files when a delay in the command processing is required to allow the NetBlazer to complete a process before continuing.

bin source [-v] <command-file>

The **source** command automatically runs a series of commands contained in the file specified by the <command-file> parameter. The -v (verbose) option displays the source lines as they are executed for debugging purposes.

Non-privileged users can only source a file that ends in .usr (e.g., **source foo.usr**). There are no restrictions, however, on **source** commands issued from a user's login command.

To use this command, you must first create a command file using a text editor. Three ways to do this are:

1. Create a command file on the NetBlazer system diskette using the **edit** command.
2. Create the command file using the text editor of an IP host system that can be reached by the NetBlazer through the network; then either use FTP to copy the file to the NetBlazer diskette or run the command file directly from remote system.

   To run a **source** command file located on a remote IP host, use **source** <host>:<filename> where <host> is the host ID or address of the remote system and <filename> is the full pathname of the file.
3. Create the command file on an MS-DOS system that supports 3.5 inch high-density diskettes, then use the MS-DOS copy utility to copy the file onto the NetBlazer diskette.

Commands in the command file must be set up to run from the Top command level and include all required parameters.

Do not use commands, like **lpdial**, that require operator input. Also, avoid using commands, such as **reboot** or **shutdown**, which may have adverse results.

The order in which the commands are entered is important. Because the commands are executed in sequence starting at the top and working down through the file, commands that define parameters used by other commands should be entered earlier in the list.

bin tee [-a] <filename>

The **tee** command sends the piped output of a command to a file that you specify in the command string and displays it on your terminal screen. You can specify a new or existing file name. If you select a new file name, the **tee** command creates that file and stores the piped output there. If you specify an existing file, the new information writes over the existing file unless you precede the file name with the -a parameter. If you include -a in your command string, the **tee** command appends the piped output to the end of the existing text in the file.

The following example shows how to use the **tee** command to save the current interface trace settings.

```
NB:Top>Configure>Interface> tracedump |tee trace.src
trace en1 none; trace en1 disabled input output headers ip off
```

Later, you can restore the current interface trace settings by entering a **source trace.src** command.

bin tcl

The **tcl** command allows system administrators, who are experienced with the UNIX Tcl command language, to set up complex command scripts.

Issuing a **tcl** command at the command prompt places the NetBlazer in interactive **tcl** command mode. A percent sign (%) is displayed in front of the command prompt to indicate that the NetBlazer is ready to accept **tcl** commands. To return the NetBlazer to normal command mode, enter a **system** or **netblazer** command.

```
NB:Top> tcl
% NB:Top> system
NB:Top>
```

To issue **tcl** commands from a script file, put a **tcl** command at the top of the file, as shown in the following example.

```
tcl
# get time of day from time command
set time [split [system time] " "]
# get hour of day
set hour [lindex [split [lindex $time 7] ":"] 0]
# get tty line number
set tty [system tty]
if {$tty=="line00" || $tty == "line01" } {
     if {$hour >= 9 && $hour < 17} { echo restricted time for $tty; logout }
     }
```

This **tcl** script restricts dial-in access to the NetBlazer between the hours of 9:00 AM and 5:00 PM.

Caution: *If you are having trouble understanding the logic in the above example, read the UNIX man pages on Tcl for information on how to use the Tcl scripting language.*

The NetBlazer supports the following standard UNIX Tcl commands: append, array, break, case, catch, concat, continue, error, eval, expr, for, foreach, format, global, history, if, incr, info, join, lappend, lindex, linsert, list, llength, lrange, lreplace, lsearch, lsort, proc, regexp, regsub, rename, return, scan, set, split, string, trace, unset, uplevel, upvar, while.

The following NetBlazer specific **tcl** commands are also provided.

echo <arguments> – sends the arguments to the terminal.

netblazer [-s] <command> <arguments> – invokes a NetBlazer command with the given arguments. Any output from the command goes to the terminal in the normal way. If -s is specified, then the return code from the command is returned to **tcl**. 0 indicates that the command was executed successfully, > 0 indicates that an error has occurred, and -1 indicates a bad usage.

system <command> <arguments> – invokes a NetBlazer command with the given arguments. Any output from the command is returned as a **tcl** result string.

ttyask <prompt> – displays the prompt and returns the resulting line typed by the user.

unknown – Any unknown **tcl** command is passed to the normal NetBlazer command processor as a **netblazer** command.

bin time

The **time** command displays the current date and time. To change the date and time, enter a **settime** command. You will be prompted for the new date and time.

bin tty

The **tty** command displays the line or port that you are currently logged in through.

```
NB:Top> tty
143.191.12.26:10935
```

To get a list of users that are currently logged into the NetBlazer, use the **who** command.

bin type <filename>

The **type** command sends the entire specified text file to your terminal screen. You can view, but not alter, the text. This command is similar to the **more** command, which sends one screenful of text at a time.

bin uptime

The **uptime** command displays the current date and time, the number of hours and minutes the NetBlazer has been running since the last time it was started, and the number of users currently logged onto the system.

bin version

The **version** command displays information about the current version of the NetBlazer software.

bin wall [<message>]

The **wall** command immediately sends a message to all users currently logged in to the NetBlazer.

You can enter a single line message or a multi-line message consisting of up to 990 characters.

The following example shows you how to enter a single line message.

```
NB:Top> wall "Hello NetBlazer users\!"
wall "Hello NetBlazer users!"

Message from root on 143.191.65.1:35361
Hello NetBlazer users!
```

When entering a single line message that contains an exclamation point, you must enter it as \! to avoid confusion with a history reference.

The next example shows you how to enter a multiline message.

```
NB:Top> wall
Enter the message for 'wall' (up to 990 characters).
Type . at the start of a line to end the message.
Line 1: The NetBlazer will be shut down 30 minutes from now.
Line 2: Please finish what you are doing and log out.
Line 3: .

Message from root on 192.124.95.18:11329
The NetBlazer will be shut down 30 minutes from now.
Please finish what you are doing and log out.
```

Note that the **shutdown** command also sends a warning message to notify all users that the NetBlazer is being shut down.

To send a messages to specific users, use the **write** command.

bin where [-n] [-c] [<args> ...]

The **where** command shows you where you in the command tree. This command is intended for use from aliases and scripts. The -n option suppresses the trailing new line. The -c option shows the commands to get to the current level. Additional arguments are echoed on the stdout.

Examples of how this command might be used is shown below:

```
echo -n "You are at the "; where "Level."
echo -n "The command to return here is: " ; where -c
```

bin who [am i]

The **who** command displays a list of users who are currently logged into the NetBlazer.

```
NB:Top> who
user       time          from
root       Dec 16 14:41  line00
root       Dec 17 14:03  143.191.12.26:10935
```

If the same account is used multiple times as shown above, and you want to know which line or port you are logged in on, you can enter a **who am i** command to get just your login information.

```
NB:Top> who am i
user       time          from
root       Dec 17 14:03  143.191.12.26:10935
```

The **write** command immediately sends a message to all users logged in with the user ID specified by the <user> parameter, unless the <location> is specified. In this case, only the user logged in from the specified location is sent the message.

For example, to send a message to all users logged in as root, issue a **write root** command as illustrated below.

```
NB login:  root

NB:Top> who
user      time          from
root      Dec 16 14:41  line00
root      Dec 18 05:50  143.191.12.26:12104
NB:Top> write root
Enter the message for 'write' (up to 990 characters).
Type . at the start of a line to end the message.
Line 1: Root users please login using your own IDs
Line 2: so that we know who you are.
Line 3: .

Message from root on 143.191.12.26:12104
Root users please login using your own IDs
so that we know who you are.
```

The next example shows you how to send a message to just the user who is logged in from a specific location.

```
NB:Top> who am i
user      time          from
root      Dec 18 05:50  143.191.12.26:12104
NB:Top> write root 143.191.12.26:12104
Enter the message for 'write' (up to 1024 characters).
Type . at the start of a line to end the message.
Line 1: This message is to myself.
Line 2: .

Message from root on 143.191.12.26:12104
This message is to myself.
```

To send a message to all logged-in users regardless of their user IDs, use the **wall** [<message>] command.

Configure Command

The **configure** command takes you to the Configure command level. To view a list of the commands available at this level, type a question mark (?) at the Configure command prompt.

```
NB:Top> configure
NB:Top>Configure> ?
Available commands:
alias>          appletalk>      arp>            asy>
boot            debug>          destination>    dialout>
ethernet        files           gmtoffset       group>
hostname        interface>      ip>             ipx>
kerberos>       line>           message>        msglog
phonelist>      ppp>            save            security>
servers>        setconsole      slip>           snmp>
syslog>         top>            up>             user>
?
```

For a description of the above configuration commands, refer to the specific command heading. For example, to read about the **dialout** commands, look under Dialout Commands in this manual.

All commands at or beneath the Configure command level are available only to users who have been given sysconf or sysstat privileges. User with sysconf privileges can use any of the commands at or below this level to configure the NetBlazer. Users with sysstat privileges are restricted to displaying information about the current configuration of the NetBlazer; they cannot change the configuration.

Caution: *The **debug** commands are provided primarily to aid Telebit in isolating problems in the NetBlazer. Do not use the **debug** commands without first contacting Telebit.*

Debug Commands

The **debug** command takes you to the Configure Debug command level. To view a list of the commands available at this level, type a question mark (?) at the Configure Debug command level.

```
NB:Top> debug
NB:Top>Configure>Debug> ?
Available commands:
alloc            args             autoconfigure    chat
clear            core             crashlog         db
devs             dh               display          dl
dump             filter>          gdt              hardware
hex              hogmemory        ident            idle
idt              inb              irq              kill
maplist          memory           memset           modemcap
noautoconfigure  nocore           notnclient       notnserver
openfiles        outb             printf           process
ps               psignal          remote           sessions
show             status           time             timers
tnclient         tnserver         top>             unmap
up>              watch            watchdog         writebyte
writelong        writeshort       ?
```

Caution: *These commands, provided to help Telebit resolve software problems, are unsupported and likely to change without notice. Because many of these commands may have undesirable effects, you should not use commands at this level unless instructed to do so by Telebit's Technical Support department.*

debug alloc [<value>]

The **debug alloc** command sets the allocation debug level, which is only available if memory allocation debugging is compiled into the NetBlazer software. The <value> parameter is a summation of any of the following options:

1	Set and check a fence value on malloc and free
2	Set memory after free
4	Set memory after alloc
8	Set watch point on fence after data
16	Set watch point on fence

Note that settings 2 and 4 degrade performance.

If the <value> parameter is not entered, the current option settings are displayed.

debug args

The **debug args** command displays the argument list.

debug autoconfigure

The **debug autoconfigure** command sends debug messages to the console from the line autoconfigure function. The **debug noautoconfigure** command stops sending line autoconfigure messages.

debug chat [<script-name>]

The **debug chat** command displays the specified chat script. If you do not enter a chat script name, this command lists of chat scripts contained in the CHAT.TXT file.

debug clear

The **debug clear** command clears any watch points.

debug core

The **debug core** command turns on core dumps. The **debug nocore** command turns off core dumps.

debug crashlog [on|off] [list] [host <ip>] [ether <hardware-addr>] [level <log-level>]

The **debug crashlog** command configures the NetBlazer to send one or more messages with the crash dump information to syslog.

By default, critical level syslog crash messages are sent to the same host specified by the **debug remote** command in the STARTUP.CNF file. If no crash dump remote host exists, the messages are sent to the broadcast address.

The **debug crashlog on|off** options turn the crashlog function on or off.

The **debug crashlog list** option displays the current crashlog settings.

The **debug crashlog host** <ip> option specifies the IP address or domain name of the syslog host.

The **debug crashlog ether** <hardware-addr> option specifies the Ethernet address of the syslog host or the gateway to reach the syslog host.

The **debug crashlog level** <log-level> option sets the syslog message severity level for syslog crash messages. Refer to the syslog command description for a list of syslog message severity levels.

If you do not specify a severity level, the NetBlazer uses the default value of critical.

debug db <address> [<count>]

The **debug db** command is the same as the **display** command, except that the memory is displayed as 8-bit hex quantities.

debug devs [verbose]

The **debug devs** command lists logical names of devices that the NetBlazer software supports. The verbose option provides additional information about the logical devices.

debug dh <address> [<count>]

The **debug dh** command is the same as the **display** command, except that the memory is displayed as 16-bit hex quantities.

debug display <address> [<count>]

The **debug display** command displays the specified memory address in 32-bit hex words and ASCII for the <count> values. Addresses that are invalid are rejected.

debug dl <address> [<count>]

The **debug dl** command is the same as the **display** command.

debug dump [reboot]

The **debug dump** command forces a core dump. This records the current state of the NetBlazer for later analysis by Telebit Technical Support. If you enter a **debug dump reboot** command, the NetBlazer automatically reboots when the core dump is completed.

Entering a control-alt-f10 on an AT keyboard also forces a dump.

debug filter

The **debug filter** command takes you to the Configure Debug Filter command level. These commands, which work the same as the standard packet filtering commands, filter the **interface trace** output for one or more interfaces. Refer to the **interface trace** and **filter** command descriptions for additional information.

debug gdt <start> [<end>]

The **debug gdt** command displays the global descriptor table entries ranging from <start> to <end>.

debug hardware

The **debug hardware** command provides information about the NetBlazer's hardware configuration.

debug hex <filename>

The **debug hex** command produces a hex dump listing of the specified file.

debug hogmemory <bytes>

The **debug hogmemory** command grabs the specified number of bytes from memory.

Caution: *This command only should be used in a test environment to force memory shortages for testing memory stress situations.*

debug ident <filename> ...

The **debug ident** command displays the revision level of the specified files.

debug idle

The **debug idle** command displays the CPU idle percentage over the last second and the number of times timerproc was skipped.

debug idt <start> [<end>]

The **debug idt** command displays the interrupt descriptor table entries that range from <start> to <end>.

debug inb <port> <count>

The **debug inb** command displays the contents of an I/O port or ports if count is specified.

Caution: *This command can be dangerous, as some ports do unpredictable things if they are read. This can bring down the system or confuse the drivers.*

debug irq <value>

The **debug irq** command displays the status of the I/O interrupt that is specified by a value between 0 and 15.

debug kill <pid>

The **debug kill** command kills most processes. You are not allowed to kill system processes (e.g., main, timer, network, tracer, or killer).

Caution: *Do not use this command unless Telebit's Technical Support department instructs you to do so.*

debug maplist <addr> <count>

The **debug maplist** command displays the memory management map for the given address and page count.

debug memory <command>

The **debug memory** command invokes memory commands that are now available as **debug** commands. For example, **debug alloc** and **memory alloc** are equivalent commands.

debug memset <addr> <char> <length>

The **debug memset** command sets the memory at <address> to the hex byte value specified by <char> for the number of bytes specified by the <length> parameter.

Caution: *Do not use this command unless Telebit's Technical Support department instructs you to do so.*

debug modemcap [<modem>]

The **debug modemcap** command displays the MODEMCAP.TXT entry for a specified modem. If you do not enter the optional <modem> parameter, the NetBlazer displays a list of modems supported by the MODEMCAP.TXT file.

debug noautoconfigure

The **debug noautoconfigure** command stops sending line autoconfigure messages to the console.

debug nocore

The **debug nocore** command turns off core dumps.

debug notnclient

The **notnclient** command turns off Telnet client option negotiation debugging.

debug notnserver

The **notnserver** command turns off Telnet server option negotiation debugging.

debug openfiles

The **debug openfiles** command displays a list of open files.

debug outb <port> <value>

The **debug outb** command sends <value> to the I/O port specified by the <port> parameter.

debug printf <format> <value ...>

The **debug printf** command displays <value ...> in the specified <format> using the tprintf routine.

debug process [<pid>]

The **debug process** command displays the information in the proc structure for the process specified by <pid>, or for the current process if you do not enter <pid>.

debug ps

The **debug ps** command invokes the **list ps** command. This command is here so that the command interpreter does not interpret **ps** as an abbreviated **psignal** command.

debug psignal <value> <count>

The **debug psignal** command wakes up the <count> processes waiting on <value>.

Caution: *Do not use this command unless Telebit's Technical Support department instructs you to do so.*

debug remote [on|off] [en0|en1|en2] [file <filename>] [host <hostid>] [ether <hardware-addr>] [myip <myip>] [list] [test]

The **debug remote** command configures the NetBlazer to send core dumps to a remote IP host through a specified Ethernet interface.

If you just turn remote dumping on and do not supply the host's hardware address and filename, the BOOTP protocol gets the required information from a BOOTP server. If there is no BOOTP server or the BOOTP server is down, then the network dump fails, and the core dump is sent to a CORE.DMP file on the NetBlazer diskette.

The actual dump is sent to a remote host by TFTP, which means that there must be a TFTP server running on the remote host. Note that TFTP requires an existing file that can be written to by everybody.

Before using the remote dump facility, verify that TFTP is functioning on the remote host.

If you wish to take dumps over the network without using BOOTP, you must enter the necessary information such as the ip address, the hardware address, and the filename with the **debug remote** commands.

The **debug remote on|off** command turns the remote core dump function on or off.

The **debug remote en0|en1|en2** command specifies which Ethernet interface to use when sending a core dump to a remote host. If you do not specify an interface, en0 is used by default.

The **debug remote host** <ip> command specifies the IP address or domain name of the TFTP server.

The **debug ether** <hardware-addr> command specifies the Ethernet address of the TFTP server or the gateway to reach the TFTP server.

The **debug remote myip** <myip> command specifies the IP address of this NetBlazer. Normally, this command is not required because the NetBlazer's global IP address is used.

The **debug remote list** command displays the current remote core dump information.

The **debug remote test** command tests the remote core dump function. The NetBlazer is disabled until the test finishes.

debug sessions

The **debug sessions** command displays the number of foreground and background sessions currently running on the NetBlazer.

debug show

The **debug show** command shows the internal console screen log, which keeps the last 4096 bytes that have been written to that screen. This is particularly useful from remote locations and when there is no external console terminal.

debug status [arguments]

The **debug status** command runs a status monitoring session on a terminal or on the internal console screen. The status display currently must be run on a terminal that supports the ANSI escape sequences.

Issuing a **debug status start** command starts a standard status monitoring session, which displays the status of all items available on this NetBlazer, as illustrated in the following example.

```
host      pearl          1.91x1-beta     time      11:04:03    date      11/19/92
          *-------------------------------------------------*  Value   %
Units
cpu       |X                                                |    6     3%
ints      |0000                                             |  205
line00    |                                                 |    0     0% C/S
line01    |                                                 |    0     0% C/S
syn0      |                                                 |    0     0% C/S
en0       |                                                 |    5     1% P/S
ibuf      |LLLLLLLLSSSSSSSSSSSSSSS                          |  239    47% #
mbuf      |                                                 |    9     1% #
memfree   |XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX                  |  765K   60%
fd        |                                                 |    0     0%
          *-------------------------------------------------*
```

The maximum values of various status fields (ints, ethernet, token ring) are nominal values that are chosen to be reasonable in most cases.

By default, the system updates this display each second and refreshes (redraws) it after every tenth update.

To stop the status monitoring session, escape out of the session (e.g., enter ctrl-] <return>), then issue a **session close** or **debug status clear** command to close the session and reset the status monitor.

The **delay** <value> and **refresh** <value> arguments control how often the system updates and redraws the status display. The delay value specifies the delay in seconds between updates, and the refresh value specifies the number of updates to display before refreshing (redrawing) the display. For example, a **debug status start delay 2 refresh 5** command starts a standard status monitoring session that is updated every two seconds and refreshed after every fifth update.

The **internal** option redirects the status monitor output to a previously installed internal video board and monitor.

The **debug status list** command displays the status monitor's current settings.

You can build a custom status display using the following command syntax:

```
debug status <item> [line <line>] [col <column>] [notitle] [maximum <value>]
```

Items that can be shown in the status display are defined in Table DEB-1.

Table DEB-1. Status Display Items

Item	Description
host	Show the host name of the NetBlazer.
time	Show the current time.
date	Show the current date.
text <string>	Display the text string. This string must be enclosed in quotes.
cpu	Show the CPU time used.
ints	Show the interrupts status.
memfree	Show the current available memory.
ibuf	Show the interrupt buffers.
mbuf	Show the mbufs.
fd	Show the floppy disk I/O.
asyXX asy <line-name>	Show the transmit and recieve status of a specified line. To display all of the asynchronous lines, use the asys command.
iface <interface> interface <interface>	Show the packets transmitted for a specified interface. To display all of the interfaces, use the nets command.
asys	Show the status for all of the NetBlazer's asynchronous lines.
nets	Show the status for all of the NetBlazer's network adapters.

Each item is displayed in the order entered. The **line** <line> and **col** <column> parameters set the position of an item.

The **notitle** option suppresses the title that is normally displayed.

The **maximum** <value> parameter specifies a value to override the item's default value.

The following screen shows the **debug status** commands used to build the standard screen.

```
NB:Top>Configure>Debug> status standard
config debug status host line 1
config debug status time line 1 col 40
config debug status date line 1 col 60
config debug status text "*-----------------------------------------------
**
col 10
config debug status text "Value  % Units" col 64 line 2
config debug status cpu
config debug status ints
config debug status asys
configure debug status line00
configure debug status line01
config debug status nets
configure debug status interface en0
configure debug status interface en1
config debug status ibuf
config debug status mbuf
config debug status mem
config debug status fd
config debug status text "*-----------------------------------------------
**
col 10
```

debug time <command>

The **debug time** command displays how long it took to execute a specified command.

debug timers

The **debug timers** command displays the current pending timers.

debug tnclient

The **debug tnclient** command turns on Telnet client option negotiation debugging.

debug tnserver

The **debug tnserver** command turns on Telnet server option negotiation debugging.

debug unmap <address> <count>

The **debug unmap** command unmaps the specified address from the address space. About the only address that can usefully be unmapped is page zero, which is already unmapped.

Caution: *Do not use this command unless Telebit's Technical Support department instructs you to do so.*

debug watch <addr> <type> [124] [WRE]

The **debug watch** command sets a watch point at the address of the given type. The type should consist of [erw][124] (access type, length).

Caution: *Do not use this command unless Telebit's Technical Support department instructs you to do so.*

debug watchdog en0|en1|en2 <delay>

The **debug watchdog** command reset the specified Ethernet interface if no input packets are within the number of seconds specified by the <delay> parameter, and output has been sent recently. This only should be necessary for UTP connections that appear to hang.

Setting the <delay> parameter to 0 turns off the watchdog for the specified Ethernet interface.

debug writebyte <address> <value>

The **debug writebyte** command writes a byte value into the specified address.

Caution: *Do not use this command unless Telebit's Technical Support department instructs you to do so.*

debug writelong <address> <value>

The **debug writelong** command writes a long value into the specified address.

Caution: *Do not use this command unless Telebit's Technical Support department instructs you to do so.*

debug writeshort <address> <value>

The **debug writeshort** command writes a short value into the specified address.

Caution: *Do not use this command unless Telebit's Technical Support department instructs you to do so.*

Files Command

The **files** command sets or displays the location of the NetBlazer's standard system and configuration files. This allows you to keep some or all of the NetBlazer standard files on a remote TCP/IP host.

files

Entering a **files** command without any parameter shows the location for all of the NetBlazer system and configuration files.

```
NB:Top> files
files          files.cnf
startup        startup.cnf
passwd         passwd
chat           chat.txt
modemcap       modemcap.txt
default        default.cnf
token          token002.mcd
overlay
bootfile       fred
end
```

Initially, all NetBlazer system and configuration files are located on the NetBlazer system diskette. However, you can arrange to have the NetBlazer use BOOTP and/or TFTP to access NetBlazer system and configuration files on a remote TCP/IP host. Refer to the Network Management chapter in the *NetBlazer Configuration Guide* for instructions on configuring the NetBlazer to use remote boot and configuration files.

The following displays shows the output of the **files** commands for a NetBlazer that has been configured to get its system and configuration files from a remote host.

```
pearl:Top> files
files          143.191.10.100:/home/NB/pearl/site.cnf
startup        143.191.10.100:/home/NB/pearl/startup.cnf
passwd         143.191.10.100:/home/NB/pearl/passwd
chat           143.191.10.100:/home/NB/chat.txt
modemcap       143.191.10.100:/home/NB/modemcap.txt
default        143.191.10.100:/home/NB/default.cnf
token          143.191.10.100:/home/NB/token002.mcd
overlay        143.191.10.100:/home/NB
bootfile       143.191.10.100:/home/NB/fred
end
```

Note: *Because the NetBlazer uses TFTP to read/write remote files, these files must have read/write permissions set for everyone.*

files <type>

Entering a **files** <type> command shows the location of the file specified by the <type> parameter.

```
pearl:Top> files startup
143.191.10.100:/home/NB/pearl/startup.cnf
```

files <type> [<host>:]<filename>

Entering a **files** <type> [<host>:]<filename> command specifies the location of the NetBlazer system and configuration files. This allows you to keep some or all of the NetBlazer standard files on a remote TCP/IP host.

The following display shows the contents of the FILES.CNF file for a NetBlazer that is configured to get its files from a TCP/IP host on the network.

```
configure file startup 143.191.10.100:/home/NB/pearl/startup.cnf
configure file passwd 143.191.10.100:/home/NB/pearl/passwd
configure file modem 143.191.10.100:/home/NB/modemcap.txt
configure file chat 143.191.10.100:/home/NB/chat.txt
configure file default 143.191.10.100:/home/NB/default.cnf
configure file token 143.191.10.100:/home/NB/token002.mcd
configure file overlay 143.191.10.100:/home/NB
```

If you have several NetBlazers on your network, you can use the %s format to indicate the host name of the NetBlazer. For example, the %s string in the following display tells the NetBlazer to substitute its host name in place of the %s string.

```
configure file startup 143.191.10.100:/home/NB/%s/startup.cnf
configure file passwd 143.191.10.100:/home/NB/%s/passwd
configure file modem 143.191.10.100:/home/NB/modemcap.txt
configure file chat 143.191.10.100:/home/NB/chat.txt
configure file default 143.191.10.100:/home/NB/default.cnf
configure file token 143.191.10.100:/home/NB/token002.mcd
configure file overlay 143.191.10.100:/home/NB
```

This allows you to use a single FILES.CNF file instead of one per NetBlazer. A NetBlazer named pearl gets its startup and password information from the files in the /home/NB/pearl directory of the host at 143.191.10.100; where as a NetBlazer named jade gets its startup and password information from the /home/NB/jade directory of the same host.

Entering a **files** <type> <command> command allows you to enter a command in place of the name of a file containing commands. This is especially useful when you want the NetBlazer to invoke a particular command after it has completed the startup routine.

The following example illustrates how a **files end** <command> could be used to configure the NetBlazer to keep the NetBlazer's time in sync with a NTP server.

```
pearl:Top> configure file end "tcl {while (1) {ntp -s napa; sleep 3600}}"
```

When the NetBlazer has finished booting, this command starts a TCL routine that updates the NetBlazer's time by querying a NTP server every hour (3600 seconds).

Gmtoffset Command

The **gmtoffset** [<hhmm>] command sets the offset in hours and minutes west of GMT. If no offset parameter is specified, the current offset is displayed.

Note: *You must set the NetBlazer's GMT offset before using* ***ntp*** *or* ***kerberos****.*

The following display shows the **gmtoffset** command used to set the GMT offset for Telebit's Sunnyvale Operations, which is in the Pacific Standard Time Zone.

```
NB:Top> gmtoffset 800
NB:Top> gmtoffset
Local time is  8:00 West of GMT.
```

During Daylight Savings Time, the GMT offset is one hour earlier. For example, the Pacific Daylight Savings Time offset is 7:00 West of GMT.

Interface Commands

The **interface** command takes you to the Configure Interface command level. To view a list of the commands available at this level, type a question mark (?) at the Configure Interface command level.

```
NB:Top> interface
NB:Top>Configure>Interface> ?
Available commands:
attach          backup          delete          dump-mib
dynamic>        list            mdelay          mru
mtu             rif             start           stop
top>            trace           tracedump       tracelimit
up>             ?
```

Commands at this level configure and control the interfaces used by the NetBlazer.

The NetBlazer uses three types of interfaces: LAN, dynamic, and dedicated.

LAN interfaces are automatically created for each Ethernet and Token Ring board installed in the NetBlazer. Before using these interfaces, you must configure the interfaces as described in the *NetBlazer Configuration Guide*.

Dynamic interfaces provide dial-up links to route packets over the public switched telephone network. Refer to the **interface dynamic** command descriptions for instructions on setting and maintaining dynamic interfaces.

Dedicated interfaces route packets over dedicated point-to-point serial links. Use the **interface attach** command to attach a dedicated interface to a synchronous or asynchronous serial line.

You can start a **watchdog** server to monitor the state of a dedicated interface. If the dedicated interface link fails, the NetBlazer automatically reroutes the link's traffic through a dynamic backup interface. A backup interface is assigned to a dedicated interface through the **interface backup** command.

interface attach <line-name> slip|ppp <interface> [<mru>] [<ip-addr>]

The **interface attach** command configures a synchronous or asynchronous serial line for routing packets over a dedicated PPP or SLIP interface. As illustrated in Figure INT-1, a dedicated PPP or SLIP interface may be used to interconnect two NetBlazers or to attach a workstation that runs PPP or SLIP software to the NetBlazer.

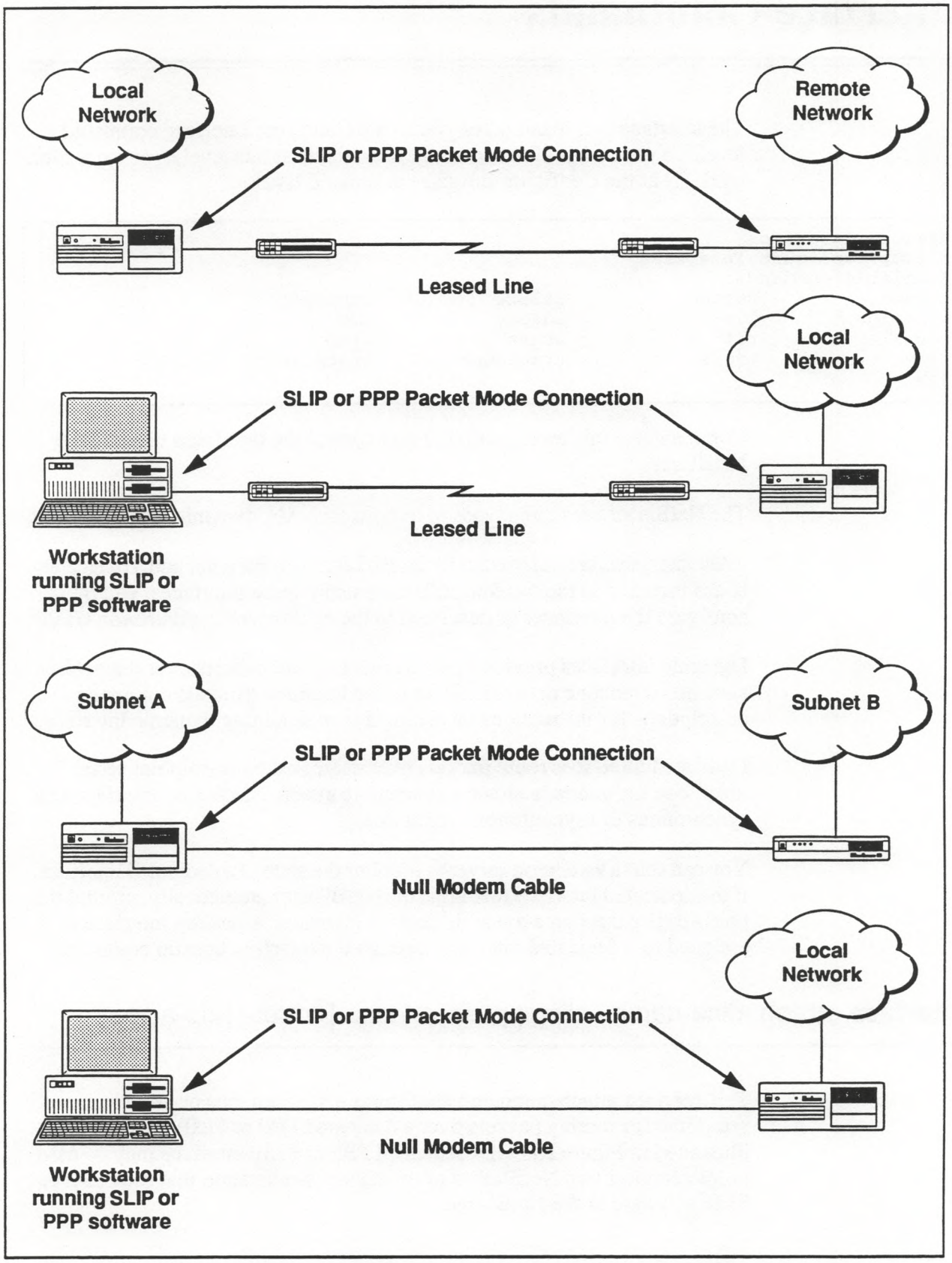

Figure INT-1. Dedicated SLIP or PPP Links

Note: *Creating a dedicated interface automatically turns off* **login** *for the line used by that interface.*

The <line-name> parameter specifies which serial line to use as a dedicated SLIP or PPP link.

The slip | ppp parameter determines the framing protocol used to transmit packets over the serial link. Synchronous lines must use PPP framing. If the interface is used for transferring only IP packets over an asynchronous link, you may choose SLIP framing. If the interface is used to transfer Appletalk or IPX packets, use PPP framing.

The <interface> parameter is the name that you assign to the new interface created by this command.

The optional <mru> (maximum receive unit) parameter sets the number of bytes in the largest packet that may be received over the dedicated link. This parameter defaults to 1006 for a SLIP link and 1500 for a PPP link.

The optional <IP-address> parameter assigns an IP address to the named interface. If this parameter is not entered, the NetBlazer global IP address is assigned to the interface.

The following example illustrates how to attach a dedicated PPP interface, named syn-opal, to the syn0 synchronous line.

```
pearl:Top>Configure>Interface> attach syn0 ppp syn-opal
syn0: interface=syn-opal, protocol=PPP, MRU=1500, ip address=143.191.63.1
```

Before you can actually use this dedicated PPP link, you must configure the NetBlazer to route packets through this interface. You also must repeat this procedure on the other side of the link. Refer to the *NetBlazer Configuration Guide* for additional instructions on setting up a dedicated link.

interface backup <interface> [<backup-interface>]

The **interface backup** command selects a dynamic interface to back up a dedicated serial line interface where <interface> is the name of the interface, and <backup-interface> is the name of the dynamic interface to use when the **watchdog** function determines that the interface line is down.

When the interface goes down and the NetBlazer switches to the backup interface, all routes using the interface are changed to the backup interface. When the interface line returns, the routes through the original interface are returned.

Entering an **interface backup** <interface> command without specifying the <backup-interface> name turns off backup for the specified interface.

interface delete <interface>

The **interface delete** command deletes a specified dynamic or dedicated interface from the interface list. You cannot delete the loopback, null, error, or static interfaces.

Note: *Any user, dialout destination, and routes associated with a dynamic interface are also deleted.*

The ***line login*** *for a attached line is not turned back on. You must turn login on if you want the login prompt to be issued when a connection is made to the line.*

interface dump-mib <interface>

The **interface dump-mip** command displays the variables maintained by each interface for SNMP as defined in RFC-1213.

interface dynamic

The **interface dynamic** command takes you to the Configure Interface Dynamic command level. To view a list of the **dynamic** commands, type a question mark (?) at the Configure Interface Dynamic command level.

```
NB:Top>Configure>Interface> dynamic
NB:Top>Configure>Interface>Dynamic> ?
Available commands:
add             auto-arp        auto-route      delete
farside         framing         ipdial          list
pppdial         prompt          queue-max       slipdial
timeout         top>            up>             ?
```

Commands at this level set up and maintain dynamic interfaces, which are used to route packets over the public telephone network.

Note: *The preferred method for creating a dynamic interface is through the* ***ipdial, slipdial,*** *or* ***pppdial*** *commands. Using these commands ensures that all of the information needed to set up a dynamic interface is provided.*

If you decide to use the ***dynamic add*** *command instead, you must create a dialout and/or user record with the same name as the interface. This allows the NetBlazer to establish outbound and/or inbound connections through the interface. You also must add one or more routes to the routing table to send packets destined for the remote site through the interface.*

Adding ARA users through the **user add** command creates a dynamic interface for ARA connections.

Refer to the *NetBlazer Configuration Guide* for detailed instructions on how to set dynamic interfaces.

interface dynamic add [<interface>] [<uname>] [<password>] [<crypto>] [<chat>] [slip|ppp|ara] [<mru>]

The **dynamic add** command creates a new dynamic interface for a dial-up connection to a remote site. Each parameter in this command must be entered in sequence. Enter 0 as a placeholder for parameters that you want set to <<NONE>>. If you omit any of the parameters, you are prompted for the remaining parameters.

Note: *Before adding a dynamic interface, first create a user record and/or dialout record with the same name as the interface. Refer to the* ***user add*** *and* ***dialout add*** *command descriptions for instructions.*

The <uname>, <password>, <crypto> and <chat> parameters are assigned to a dialout record with the same name as the <interface>. If the dialout record does not exist, these parameters are set to <<NONE>>.

The <interface> parameter is the name assigned to the interface.To allow inbound or outbound connections through this interface, you must set up a user record and/or dialout record with the same name as the interface.

The <uname> parameter is the user ID that is sent in response to a login prompt from the remote NetBlazer. The <password> parameter is sent in response to a password prompt, and the <crypto-key> parameter is used to encrypt a crypto response when a crypto challenge is received from the remote site.

To authenticate a dial-out connection, the remote NetBlazer must have a user record that matches the interface's <uname>, <password>, and <crypto-key> parameters.

The <chat> parameter specifies which chat script in the CHAT.TXT file to use when attempting to establish a dial-out connection with the remote site. Use the ics chat script for interfaces used to dial out to another NetBlazer.

The slip|ppp|ara parameter specifies which protocol to use when sending packets through this interface. If the interface is used to transfer IPX or AppleTalk packets, use PPP. If the interface is used for an ARA connection, choose ARA. If the interface is used to transfer only IP packets, you may use either PPP or SLIP.

The <mru> parameter sets the number of bytes in the largest packet that may be received or transmitted through this interface. We recommend that you use the default MRU/MTU setting: 1500 for PPP or ARA connections, 1006 for SLIP connections.

interface dynamic auto-arp <interface> <add|delete> <ip-addr ...>

The **dynamic auto-arp** command configures the NetBlazer to automatically create proxy ARPs and related routes for one or more host IP addresses when a connection is established through a specified dynamic interface. This accommodates users who temporarily need to move their system from the local network to a remote site.

Note: *Using the auto-arp function limits establishment of a dynamic connection to the inbound direction only. In other words, hosts on the local side cannot communicate with the remote site through this interface until a connection is established from the remote site.*

Refer to the *NetBlazer Configuration Guide* for an example of how the **dynamic auto-arp** command is used.

The **dynamic auto-arp** <interface> **add** <ip-addr ...> command adds host IP addresses to the list of addresses for which the NetBlazer creates proxy ARPs and related routes when the specified dynamic interface is connected. The auto-ARP entries (shown as **published,auto** in the ARP list) and related routes (marked by an A in the P field of the route list) only remain while the connection is present. When the connection is dropped, the auto-ARP entries and routes are removed.

The **dynamic auto-arp** <interface> **delete** <ip-addr ...> command removes host IP addresses from the list of addresses for which the NetBlazer creates proxy ARPs and related routes when a dynamic connection is made.

interface dynamic auto-route <interface> <add|delete> <ip-addr[/bits] ...>

The **dynamic auto-route** command configures the NetBlazer to automatically add one or more host or network IP addresses to the routing table when a connection is established through a specified dynamic interface. When the connection is dropped, the dynamic interface's routes, marked by the letter A in the P field of the route list, are removed. This reduces the number of entries in the routing table by only listing routes through a dynamic interface when it is in use.

Note: *Using the auto-route function limits establishment of a dynamic connection to the inbound direction only. In other words, hosts on the local side cannot communicate with the remote site through this interface until a connection is established from the remote site.*

The **dynamic auto-route** <interface> **add** <ip-addr[/bits] ...> command adds IP addresses to the list of auto-route addresses for a specified interface.

The **dynamic auto-route** <interface> **delete** <ip-addr[/bits] ...> command removes IP addresses from the auto-route address list for the specified interface.

Refer to the *NetBlazer Configuration Guide* for an example of how the **dynamic auto-route** command is used.

interface dynamic delete <interface>

The **dynamic delete** command removes a specified dynamic interface from the interface list.

Note: *Deleting a dynamic interface also removes any user and/or dialout record with the same name as the interface and purges the routing table of any routes through that interface.*

interface dynamic farside <interface>

The **dynamic farside** command displays information describing how a connection to the remote site is established through this interface. This information is also displayed at the end of an **ipdial** routine.

interface dynamic framing <interface> slip|ppp

The **dynamic framing** command selects the framing method used to transfer packets through a specified dynamic interface. The same framing method must be used on both sides of a dynamic link. If the interface is used for transferring only IP packets, choose SLIP framing. If the interface is used to transfer AppleTalk or IPX packets, use PPP framing.

interface dynamic ipdial

The **dynamic ipdial** command is one of three routines that step you through the process of creating a dynamic dial-up interface. The **ipdial** routine is the best choice for setting up multi-line connections to the same destination. If you are setting up a single line interface, consider using **slipdial** or **pppdial** instead. These two routines simplify the process by using the NetBlazer's defaults whenever possible.

You need the following information to set up a dynamic interface using **ipdial**:

- The user IDs, passwords, and crypto keys used to authenticate an inbound and/or outbound connection.

- The telephone number(s) used to call the remote NetBlazer.
- The serial link framing protocol (SLIP or PPP) used to encapsulate IP packets sent between the two sites. Both sites must use the same framing method. If the interface is used for transferring only IP packets, choose SLIP framing. If the interface is used to transfer AppleTalk or IPX packets, use PPP framing.
- The network address of each remote LAN segment that can be reached either directly or indirectly through this interface.

The same information is required at the remote site to set up the dynamic interface on the other side of the link.

Refer to the *NetBlazer Configuration Guide* for detailed instructions on how to set up dynamic interfaces.

interface dynamic list [<interface>]

The **dynamic list** command displays status information about the dynamic interfaces that are currently used by the NetBlazer.

If the optional <interface> parameter is entered, additional information for the named interface is displayed. This information varies depending on the type of interface and its present status.

interface dynamic pppdial [<interface>]

The **dynamic pppdial** command starts a dialog that steps you through the process of setting up a dynamic PPP interface to a remote site. In order to keep the dialog as simple as possible, the NetBlazer's defaults are used whenever possible.

To set up a dynamic interface using **pppdial**, you need the following information:

- The host name and telephone number of the remote NetBlazer.
- A crypto key or password used to authenticate the connection between the local and remote site.
- The network address of each remote LAN segment that can be reached through this dynamic interface.

The same information is required at the remote site to set up the dynamic interface on the other side of the link.

If you enter the optional <interface> parameter, the NetBlazer will not prompt you for the name of the remote system in the **pppdial** dialog.

Note: *When using **pppdial** to set up a dynamic interface between two NetBlazers, be sure to use the hostname of the remote NetBlazer as the interface name. Also, be sure to use the same crypto key or password for both interfaces.*

An example of the **pppdial** dialog is shown in the following display.

```
jade:Top>Configure>Interface>Dynamic> pppdial
Name of the remote (other) system: ruby
Phone number: 9-555-1234
Authentication method (crypto, password, none) [c]:
Enter the crypto key (no echo):
Re-enter the crypto key:
Add dialout 'ruby'
Add user 'ruby'
Add dynamic interface=ruby, user=jade chat=ics, with crypto handshaking
 using PPP, MRU=1500
Okay (yes|no|quit) [y]?
REMINDER! You need to set up route(s) for this interface
```

This dialog creates an interface, dialout, and user record with the same name as the remote system (ruby in the above example).

The interface record is configured to use PPP framing and an MRU/MTU of 1500 by default.

After creating the dynamic interface to the remote site, you must add the interface to the IPX and/or AppleTalk interface list (if applicable) and set up static routes through this interface for each LAN segment on the remote network.

This process of creating an interface and setting up routes must be repeated at the remote site. If the remote NetBlazer is installed and ready to accept calls, you can use the **dial** command to call the remote NetBlazer and run **pppdial** from your local NetBlazer.

interface dynamic prompt <interface> [<text-string>]

The **dynamic prompt** command sets the text string that is sent by NetBlazer when a packet mode connection is successfully established through the specified interface. Be sure to enclose <text-strings> that include spaces in quotes.

If you do not enter a <text-strings>, the NetBlazer displays the interface's current prompt setting.

For NetBlazer to NetBlazer connections, use the default prompt setting – Packet mode enabled.

interface dynamic queue-max <interface> <queue-length>

The **dynamic queue-max** command sets the maximum number of packets that may be queued for a designated interface.

Once the queue reaches the value specified by the <queue-length> parameter, additional packets are discarded.

interface dynamic slipdial [<interface>]

The **dynamic slipdial** command starts a dialog that steps you through the process of setting up a dynamic SLIP interface to a remote site. In order to keep the dialog as simple as possible, the NetBlazer's defaults are used whenever possible.

To set up a dynamic interface using **slipdial**, you need the following information:

- The host name and telephone number of the remote NetBlazer.
- A crypto key or password used to authenticate the connection between the local and remote site.
- The network address of each remote LAN segment that can be reached through this dynamic interface.

The same information is required at the remote site to set up the dynamic interface on the other side of the link.

If you enter the optional <interface> parameter, the NetBlazer will not prompt you for the name of the remote system in the **slipdial** dialog.

Note: *When using **slipdial** to set up a dynamic interface between two NetBlazers, be sure to use the hostname of the remote NetBlazer as the interface name. Also, be sure to use the same crypto key or password for both interfaces.*

An example of the **slipdial** dialog is shown in the following display.

```
jade:Top>Configure>Interface>Dynamic> slipdial
Name of the remote (other) system: ruby
Phone number: 9-555-1234
Authentication method (crypto, password, none) [c]:
Enter the crypto key (no echo):
Re-enter the crypto key:
Add dialout 'ruby'
Add user 'ruby'
Add dynamic interface=ruby, user=jade chat=ics, with crypto handshaking
 using SLIP, MRU=1006
Okay (yes|no|quit) [y]?
REMINDER! You need to set up route(s) for this interface
```

This dialog creates an interface, dialout, and user record with the same name as the remote system (ruby in the above example).

The interface record is configured to use SLIP framing and an MRU/MTU of 1006 by default.

After creating the dynamic interface to the remote site, you must add the interface to the IPX and/or AppleTalk interface list (if applicable) and set up static routes through this interface for each LAN segment on the remote network.

This process of creating an interface and setting up routes must be repeated at the remote site. If the remote NetBlazer is installed and ready to accept calls, you can use the **dial** command to call the remote NetBlazer and run **slipdial** from your local NetBlazer.

interface dynamic timeout <interface> <timeout-seconds>

The **dynamic timeout** command sets the amount of time in seconds that a NetBlazer maintains an inactive packet mode connection associated with a designated interface.

The NetBlazer starts a timer on each packet mode connection. This timer is reset each time a packet is transmitted or received through the interface. When the interface has been inactive (no packets have been transmitted or received) for the number of seconds specified by the <timeout-secs> value, the NetBlazer drops the connection.

Setting an interface's timeout value to 0 disables the timer on one side of the connection. If you set the timers on both sides of the connection to 0, the connection will never time out. In this case, you must enter a **line reset** command to break the connection.

interface list [<interface>]

The **interface list** command displays status information about the interfaces that are currently used by the NetBlazer.

If the optional <interface> parameter is entered, additional information for the named interface is displayed. This information varies depending on the type of interface and its present status.

interface mdelay <interface> [<seconds>] [<threshold>]

The **interface mdelay** command sets the parameters that the NetBlazer uses to measure the traffic through an interface in order to determine if the other dynamic interfaces in a balanced route should be brought up. Because there is no way to predict which interface in a balanced route will be used first, you should set the **mdelay** value for all interfaces in the group.

If the percentage of interface's total traffic divided by its expected capacity for the specified number of seconds exceeds the threshold value, the NetBlazer brings up the remaining dynamic interfaces in the balanced route.

The <seconds> parameter affects the NetBlazer response to instantaneous bursts in the interface's traffic – the lower this parameter is set, the more likely the NetBlazer is to bring up the remaining interfaces in response to a short burst of traffic through the interface. If you set this parameter to a higher value, the NetBlazer is more likely to bring up the other interfaces based on the average level of traffic through the interface.

If the interface's **mdelay** is set to its default value of 0 seconds, all interfaces in the balanced route are brought up immediately.

The <threshold> parameter is a percentage value that sets the trigger point at which the NetBlazer brings up the remaining interfaces. The NetBlazer is less likely to bring up the other interfaces when this parameter is set to a higher value.

interface mru [<interface>] [<mru>]

The **interface mru** command sets the maximum receive unit (i.e., number of octets or bytes) that may be transmitted through a specified interface.

If this command is entered without the <interface> or <mru> parameters, the system prompts you for the required information.

We recommend that you use the default MRU setting: 1500 for PPP or 1006 for SLIP connections.

interface mtu [<interface>] [<mtu>]

The **interface mtu** command sets the maximum transmit unit (i.e., number of octets or bytes) that may be transmitted through a specified interface.

If this command is entered without the <interface> and <mtu> parameters, the system prompts you for the required information.

You cannot set the MTU for a PPP interface. This value is negotiated based on the peer MRU.

We recommend that you use the default MTU setting of 1006 for SLIP connections.

interface rif <interface> [<node-address>]

The **interface rif** command displays the contents of the Routing Information Fields, which are currently used only in Token Ring environments. For additional information, refer to *RFC 1042, IEEE 802.5 (Token Ring Media Access Control)* or the *IBM Token-Ring Network Architecture Reference Manual*.

interface start <interface>

The **interface start** command restarts an interface that has been previously turned off by a **interface stop** command.

interface stop <interface>

The **interface stop** command turns off a specified interface. This command only works for static or dedicated interfaces. It cannot be used to stop a dynamic interface.

interface trace [<interface>] [<options ...>]

The **interface trace** command allows you to selectively monitor inbound and/or outbound traffic through one or more interfaces. Trace output for all interfaces is sent to only one serial port or telnet session – the last one used to issue a **trace** command. However, you can use the **ofile=**<filename> option to redirect the trace output to a file. For example, **trace en0 on ofile=trace.out** sends the trace output for en0 to the file named TRACE.OUT.

To view a trace file, turn off tracing, then enter a **more** <filename> command.

The <interface> parameter specifies which interface is to be traced. This is the name assigned to a selected interface.

The <options ...> parameters defines the form and content of the trace output. Refer to Table INT-1.

Table INT-1. Interface Trace Options

Option	Description
on, all	Enable input and output packet tracing for all packet types (ip, ipx, appletalk, unknown, compressed, uncompressed, raw).
dis[able], off	Stop tracing specified interface.
ena[ble], resume	Resume tracing specified interface.
clear, none	Clear all tracing option bits of specified interface.
in[put], noin[put]	Enable/disable tracing of inbound packets.
out[put], noout[put]	Enable/disable tracing of outbound packets.
comp[ressed]	Enable tracing of compressed packets.
nocomp[ressed]	Disable tracing of compressed packets.
uncomp[ressed]	Enable tracing of uncompressed packets.
nouncomp[ressed]	Disable tracing of uncompressed packets.
raw	Enable tracing of raw packets.
noraw	Disable tracing of raw packets.
ip, noip	Enable/disable IP packet tracing.
ipx, noipx	Enable/disable IPX packet tracing.
atk, noatk	Enable/disable AppleTalk packet tracing.
appletalk, noappletalk	Same as atk, noatk.
unknown, nounknown	Enable/disable tracing of unknown packet types.
onlyip	Disable packet tracing for all protocols except IP.
onlyipx	Disable packet tracing for all protocols except IPX.
onlyatk	Disable packet tracing for all protocols except AppleTalk.
onlyunknown	Disable packet tracing for all protocols except unknown packet types.
hex	Show packet contents in both HEX and ASCII format.
ascii	Show packet contents in ASCII format only.
header[s]	Only show packet headers. This is the default if HEX or ASCII is not set.

Notes:

Entering a **trace off** command without specifying an interface, stops tracing for all interfaces.

To trace a SLIP interface, you must specify at least one of the comp, uncomp or raw options. These optons are only valid when tracing SLIP interfaces.

Entering an **interface trace** command without any parameters displays the current status of tracing for each interface.

If the trace output falls too far behind (i.e., more than 200 packets in the queue), trace information is discarded to avoid tying up too much memory. If **syslog** is enabled, an informational message is logged indicating how many trace packets were lost.

You can filter trace information using the **debug filter** commands, which work the same as the standard packet filtering commands. For example, if you only want to view trace information on the en0 interface for packets that have a source or destination address of 143.191.64.1, enter the following **debug filter** and **interface trace** commands.

```
NB:Top> debug filter
NB:Top>Configure>Debug>Filter> permit 143.191.64.1 en0
NB:Top>Configure>Debug>Filter> list
Source              Destination         Interface Protocol I/O  Permit/Deny   Use
143.191.64.1/32                         en0                In   Permit          0
                    143.191.64.1/32     en0                In   Permit          0
143.191.64.1/32                         en0                Out  Permit          0
                    143.191.64.1/32     en0                Out  Permit          0
Default                                 en0                     Deny
notify on
NB:Top>Configure>Debug>Filter> interface trace en0 on
en0 enabled compressed uncompressed raw input output status ip ipx appletalk
unknown
Tracing to line line17
```

If you start a trace from a telnet session that uses the same interface being traced, debug filters are automatically set up to filter out the telnet session's packets. An example of the filters that are automatically added to the debug filter list is shown below.

```
NB:Top>Configure>Interface> debug filter list
Source              Destination         Interface Protocol I/O  Permit/Deny   Use
143.191.10.100/32   143.191.11.5/32     en0       TCP      In   Deny           12
                                          port rules: =23
143.191.11.5/32     143.191.10.100/32   en0       TCP      Out  Deny           11
                                          port rules: =4004
Default                                 en0                     Permit
notify on
```

These filters are automatically removed when the telnet session is closed.

Caution: *Do not delete the telnet session filters while tracing is active. This causes traffic to increase exponentially, rendering the connection unusable.*

interface tracedump [<interface>]

The **interface tracedump** command outputs a set of commands that, if executed, resets the interface's trace options to their current settings. If the <interface> is not specified, **tracedump** outputs commands for all interfaces that are not clear.

You can put the current trace settings in a file by piping **tracedump** output to the **output** or **tee** command as shown in the following example.

```
NB:Top> tracedump en1 |output trace.cnf
```

To later restore the trace settings saved in the TRACE.CNF file, enter a **source trace.cnf** command.

```
NB:Top> source trace.cnf
Clearing trace flags, as options would produce no output
en1 disabled
en1 disabled compressed uncompressed raw input output status ip
```

Notice in the above example that the trace options are set, but trace is not enabled until you enter a **trace** <interface> **resume** command.

interface tracelimit [<interface>] [<limit>]

The **interface tracelimit** command sets the maximum number of packets that are traced for a specified interface. Each time a packet is traced, the interface's trace limit is decremented by 1. When the interface's limit reaches 0, tracing for that interface is disabled.

Setting an interface's trace limit to 0 allows unlimited tracing. Therefore, if you resume tracing an interface that was disabled by the trace limit function, tracing for that interface is unlimited, unless you reset the interface's trace limit to a non-zero value.

Entering a **tracelimit** <interface> command without specifying the <limit> value displays the interface's current trace limit setting. Entering **tracelimit** with no parameters displays the trace limits for all interfaces that have a non-zero trace limit value.

IPX Commands

The **ipx** command takes you to the Configure IPX command level. To view a list of the commands available at this level, type a question mark (**?**) at the Configure IPX command level prompt.

```
NB:Top> ipx
NB:Top>Configure>IPX> ?
Available commands:
acs>            add             address         bang
delete          frames          ifconfig        internal-net
iptunnel>       keepalive>      list            log
netbios         netstat         ping            rip>
route>          sap>            top>            up>
wan>            ?
```

Commands at this level are used to configure and manage the IPX (Internet Packet eXchange) facilities. Commands followed by an angle bracket (>) in the above list take you to a subcommand level. For example, if you enter **route** at this level, the NetBlazer takes you to the Configure IPX Route command level.

ipx acs

The **ipx acs** command takes you to the Configure IPX ACS command level. To view a list of the commands available at this level, type a question mark (**?**) at the Configure IPX ACS command level prompt.

```
NB:Top>Configure>IPX> acs
NB:Top>Configure>IPX>ACS> ?
Available commands:
audit           deregister      list            lock
start           stop            top>            up>
?
```

Use the commands at this level to configure and manage the NetBlazer Asynchronous Communications Server (ACS). NetBlazer ACS is an option you can purchase separately.

ipx acs audit [on|off]

The **ipx acs audit** command turns on or off the NetBlazer ACS audit log facility. When the audit log is on, the NetBlazer sends audit log messages to the NetBlazer syslog utility. The default is off.

ipx acs deregister <user>

The **ipx acs deregister** command removes a workstation from the list of workstations registered at the NetBlazer. Normally, you do not use this command. If a workstation is removed from the network, you can use this command to remove the workstation from the list of registered workstations.

ipx acs list

The **ipx acs list** command displays a list of connection information and the list of registered workstations. The connection information is similar to that displayed by the NetBlazer ACS's management console utility, PDVIEW.EXE. This utility is supplied on a separate diskette with other NetBlazer ACS programs.

ipx acs lock <line-name> [on|off]

The **ipx acs lock** command tells the NetBlazer to lock the configuration for a line. Once the line configuration is locked, a terminal emulator cannot change line speed or flow control. However, you can change a line configuration by using the NetBlazer line commands.

The <line-name> parameter specifies which asynchronous line's configuration to lock. Examples are line01 and line13.

ipx acs start

The **ipx acs start** command starts NetBlazer ACS.

ipx acs stop

The **ipx acs stop** command stops NetBlazer ACS.

ipx add <interface>|all

The **ipx add** command sets up an existing NetBlazer interface for use by IPX.

The <interface> parameter specifies the name of the interface to which packets are routed.

The all parameter lets you set all currently available interfaces for use by IPX. Use the **configure interface list** command to see a list of currently available interfaces.

Note: *SLIP interfaces are not available to IPX.*

The first of the following sample commands adds an Ethernet interface. The second command is an example of adding a dynamic interface.

```
NB:Top>Configure>IPX> add en0
NB:Top>Configure>IPX> add ruby
```

ipx address [<ipx-addr> [<interface>]]

The **ipx address** command sets the global IPX address assigned to the NetBlazer or sets the IPX address assigned to a specified interface or line. Work with your network administrator in assigning IPX addresses.

An IPX address has three parts, and each part is identified by a maximum number of hexadecimal characters, as shown in Table IPX-1:

Table IPX-1. Parts of an IPX Address

Part	Maximum Number of Hexadecimal Characters
Network Number	8
Host Number	12
Socket Number	4

For the NetBlazer and most interfaces, enter only the network number (for example, 331) or both the network and host numbers separated by a period (for example, 300.10). Let the NetBlazer or NetWare assign the socket (or port) number unless you are assigning the address to a process that gets a well-known socket number.Some standard socket numbers are shown in Table IPX-2; numbers are hexadecimal:

Table IPX-2. IPX Socket Numbers

Socket Number	Meaning
0001h	Xerox Network Standard (XNS) RIP
0002h	Echo protocol packet
0003h	Error handler packet
0020h-003Fh	Experimental

Table IPX-2. IPX Socket Numbers (Continued)

Socket Number	Meaning
0001h-0BB8h	Registered with Xerox
0BB9h-	Dynamically assigned
0451h	NetWare file service packet (NCP)
0452h	NetWare service advertising packet
0453h	NetWare RIP
0455h	NetBios packet
0456h	diagnostic packet
4001h	NetWare watchdog packet
4003h	NetWare shell packet
4004h-7FFFh	Dynamically assigned sockets
8000h-	Socket numbers assigned by Novell

The optional <interface> parameter specifies the name of the interface to which packets are sent and through which the packets are routed.

If you enter this command without any parameters, the NetBlazer displays its global IPX address.

```
NB:Top>Configure>IPX> address
Global IPX address is 0
```

The NetBlazer is shipped with the global IPX address set to 0. The next example illustrates how to set the NetBlazer's global IPX address:

```
NB:Top>Configure>IPX> address 300
Global IPX address is 300
```

The global IPX address is automatically assigned to each interface.

Each Ethernet and Token Ring interface must have a unique IPX address. Use an **ipx address** <ipx-addr> <interface> command to set the IPX address of an interface to an address that is different from the NetBlazer's global IPX address. This example shows how to set up the IPX address for a specific Ethernet interface:

```
NB:Top>Configure>IPX> address 17 en1
Set IPX address of interface 'en1'
```

Make sure the IPX network number you give each Ethernet and Token Ring interface matches the network's IPX address for that interface.

ipx bang <ipx-addr> [<length> [<count> [<interval> [<outstanding>]]]]

The **ipx bang** command sends IPX echo requests to a specified remote NetBlazer. This command is similar to an **ipx ping** command except **ipx bang** prints only a single character (!) instead of a line of statistics for each response. Lost or out-of-sequence packets are shown as a dot (.). IPX messages are shown as an X in the **bang** display. Refer to the **ipx ping** command for an explanation of the parameters used by both commands.

ipx delete <interface>|all

The **ipx delete** command removes either one IPX interface or all IPX interfaces, depending on which parameter, <interface> or all, you enter. This command also deletes all IPX routes associated with the interface(s).

ipx frames <interface> [ethernet_802.3|ethernet_2|token-ring|token-ring _snap]

The **ipx frames** command specifies framing for Ethernet and Token Ring interfaces. The default for Ethernet is Ethernet 802.3. The default for Token Ring is Token Ring Snap.

If you use Ethernet II packets, use the **frames** <interface> **ethernet_2** command to specify the correct frames.

If you have Token Ring and you have not changed the file server to Token Ring Snap, use the **frames** <interface> **token-ring** command to specify the correct frames.

ipx ifconfig <interface> [<address>] [up|down]

The **ipx ifconfig** command provides another way to set the IPX address of an interface. You cannot set the global IPX address with the **ipx ifconfig** command, so use the **ipx address** command instead of the **ipx ifconfig** command.

The <interface> parameter specifies the name of the interface to which packets are sent and through which the packets are routed.

The optional <address> parameter is the interface's address.

The optional down parameter disables an interface. The optional up parameter enables an interface after it was brought down by an **ipx ifconfig down** command. You can bring down an interface only if you have brought it up or added it.

ipx internal-net <ipx-network-number>

The **ipx internal-net** command sets the internal network number for the NetBlazer router. You need this for NetBlazer ACS.

Work with your network administrator in assigning an internal network number. An internal network number can be up to eight hexadecimal characters. See Table IPX-1 for the parts of an IPX address. Each internal network number must be different from any other network number or internal network number. NetWare 3.11 assigns a node (or host) number of 1 to the internal network number.

ipx iptunnel

The **ipx iptunnel** command takes you to the Configure IPX IpTunnel command level. To view a list of the commands available at this level, type a question mark (**?**) at the Configure IPX IpTunnel command level prompt.

```
NB:Top> ipx iptunnel
NB:Top>Configure>IPX>IpTunnel> ?
Available commands:
add             delete          flush           list
route           top>            up>             ?
```

Commands at this level are used to configure and manage IP (Internet Protocol) tunneling. The command **iptunnel** provides a means of encapsulating IPX datagrams within IP packets so IPX traffic can travel across an IP internetwork. (An IPX datagram is carried in the data portion of an IP packet.) This allows IPX networks to be connected across internetworks that carry only IP traffic.

ipx iptunnel add <ip-addr>

The **ipx iptunnel add** command adds a peer IP address. To use an IP internetwork, the NetBlazer keeps a list of the IP addresses of other IPX routers on the IP internetwork. This is the peer list and each address on the list is a peer IP address. Make this list with the **ipx iptunnel add** command. The peer list is the list of hosts to which the NetBlazer sends IPX broadcast packets such as Routing Information Protocol (RIP) and Service Advertising Protocol (SAP) packets.

ipx iptunnel delete <ip-addr>

The **ipx iptunnel delete** command deletes a peer IP address from the peer list.

ipx iptunnel flush

The **ipx iptunnel flush** command removes all peer IP addresses from the peer list.

ipx iptunnel list

The **ipx iptunnel list** command displays the list of all peer IP addresses.

ipx iptunnel route <ipx-dest> <ip-gateway> [<metric>]

The **ipx iptunnel route** command adds an IPX iptunnel route through an IP gateway to a destination IPX network.

The <ipx-dest> parameter identifies the IPX destination, that is, the destination on a remote IPX network. An example is the IPX address of a file server on the destination IPX network.

The <ip-gateway> parameter specifies the IP address of the gateway to which the IPX packets should be sent, for example, the IP address of a NetBlazer on the destination IPX network.

The optional <metric> parameter specifies the number of hops required to reach the destination IPX network. The default value for <metric> is 1.

ipx keepalive

The **ipx keepalive** command takes you to the Configure IPX Keepalive command level. To view a list of the commands available at this level, type a question mark (**?**) at the Configure IPX Keepalive command level prompt.

```
NB:Top> ipx keepalive
NB:Top>Configure>IPX>Keepalive> ?
Available commands:
list            notify          start           stop
timeout         top>            up>             ?
```

Use commands at this level to manage IPX keep-alive messages and responses.

The NetWare watchdog timer checks idle workstations and keeps their connections with the file server "alive" as long as the workstation is running.

People using NetWare log in to servers but often continue to do most of their work on their local drives. They look to the remote server for information infrequently. When the NetBlazer makes a connection, a server sends keep-alive messages to the workstation to make sure it is still active. It would be wasteful to keep a dynamic interface active just to transfer these keep-alive messages and responses. This can be prevented with the **ipx keepalive** command.

With the **ipx keepalive** command, the remote NetBlazer (the one at the server's end) pretends to be the remote workstation and responds to keep-alive packets. The local NetBlazer (the one at the workstation end) monitors the workstation and sends keep-alive packets to it as long as it is active. If the workstation changes to the server's drive, the dial-up link is reestablished. If the workstation is down, the local NetBlazer notifies the server that the workstation is down.

You can set the idle time with the NetBlazer command **interface dynamic timeout** (make sure the idle timer is not less than 360 seconds). The IPX keep-alive packets do not update the idle timer. If a line has no other traffic than IPX keep-alive packets, RIP packets, and SAP packets, the line is considered to be idle. If a line is idle longer than the maximum idle time, the dial link is taken down.

ipx keepalive list

The **ipx keepalive list** command displays the list of keep-alive entries. Keep-alive entries are the file servers and workstations with which the NetBlazer is maintaining a logical connection with keep-alive packets.

ipx keepalive notify [on|off]

The **ipx keepalive notify** command is turned on when the **IPX keepalive** command is started. With this command on, the NetBlazer notifies the remote NetBlazer on the dial-up link of the status of keep-alive entries.

The command also displays the list of keep-alive entries.

You can use the command to turn off keep-alive notification. The NetBlazer is shipped with keep-alive notification turned on.

ipx keepalive start

The **ipx keepalive start** command starts IPX keep-alive packets. The NetBlazer is shipped with keep-alive started.

ipx keepalive stop

The **ipx keepalive stop** command stops IPX keep-alive packets.

ipx keepalive timeout [<max-time(minutes)>]

The **ipx keepalive timeout** command sets the maximum time, in minutes, for sending IPX keep-alive packets. The NetBlazer rounds up the entered value to the nearest 15 minutes. When the time is reached, the NetBlazer stops sending IPX keep-alive packets. The default specifies no limit.

ipx list

The **ipx list** command displays status information about the IPX interfaces currently used by the NetBlazer router.

ipx log [<facility> [<level>]]

The **ipx log** command is a debugging utility that sends error and debugging messages to the syslog utility.

The **ipx log** command without parameters displays a table showing activity for ACS, Net, RIP, and SAP. Net covers IPX and PPP interfaces.

The optional <facility> parameter is one of these: **acs, net, rip**, or **sap**.

The optional <level> parameter has the range 0 through 7 and specifies the level of tracing. The higher the number, the more trace information you receive. The default level is 3 (this means only errors go to the syslog).

ipx netbios <interface> [on|off]

The **ipx netbios** command controls the forwarding of netbios broadcasts. Netbios clients use netbios broadcasts to locate netbios servers. IPX routers forward these broadcasts across interfaces.

By default, forwarding of netbios broadcasts is on for all interfaces. Use the **ipx netbios** command to disable the forwarding of netbios broadcasts on an interface.

The <interface> parameter specifies the name of an interface over which netbios broadcast forwarding is enabled or disabled.

Netbios broadcasts are not forwarded over a dynamic link if the link is down. Netbios broadcasts are "spoofed", that is, the dynamic link that has only netbios broadcasts triggers the idle timer and brings the link down.

ipx netstat [-Aalimrs] | [-p ipx|spx]

The **ipx netstat** command displays a table of running statistics for packet traffic on configured network interfaces. The table shows cumulative statistics for packets transferred, errors, collisions, the network addresses for the interface, and the maximum transmission units (mtu).

Note: *The tables can change while netstat is examining them, creating incorrect or partial displays.*

The parameters work as described in Table IPX-3.

Table IPX-3. Netstat Parameters

Parameter	Function
-A	Shows address of protocol control blocks related to sockets.
-a	Shows the status of all sockets.
-l	Shows information for local routes (use with -r parameter).
-i	Shows the condition of interfaces.
-m	Shows statistics for an IPX network's private buffer pool.
-p ipx|spx	Selects protocol.
-r	Shows routing table (with -s, shows routing statistics instead).
-s	Shows protocol statistics (with -r, shows routing statistics instead).

ipx ping <ipx-addr> [<length> [<count> [<interval> [<outstanding>]]]]

The **ipx ping** command sends IPX echo requests to a specified remote NetBlazer router. A NetBlazer pings only another NetBlazer. If the first attempt to ping is not successful, enter **^C** (control-C) and ping the NetBlazer a second time.

The <ipx-addr> parameter is the IPX address of the remote NetBlazer router.

The optional <length> parameter is used to pad the request to a larger size specified in bytes. This parameter must be greater than 32 bytes (the minimum length allowed) to take effect. However, a value less than 32 can be used as placeholder in the command. The default length is 64 bytes.

The optional <count> parameter specifies the number of IPX requests to send to the remote host. If you do not enter this parameter or if you enter 0 as a placeholder, the NetBlazer continues issuing requests to the host until you stop the ping session by pressing ctrl-C.

The optional <interval> parameter specifies in milliseconds the time to wait between sending IPX echo requests. If you can set this value to 0, the NetBlazer sends pings with 10 milliseconds separation (approximately 100 pings per second). The default interval value is 0 for 10 milliseconds.

The optional <outstanding> parameter specifies the number of echo requests (pings) that may be transmitted before the NetBlazer stops and waits for their responses. The default specifies no limit.

The ability to set the ping interval to 0 (10 milliseconds between pings) and set the number of outstanding pings is a useful tool for placing a significant load on a network without overloading the network.

ipx rip

The **ipx rip** command takes you to the Configure IPX RIP command level. (The **ipx rip** command is not the same as the **ip rip** command.)

To view a list of the commands available at this level, type a question mark (**?**) at the Configure IPX RIP command level prompt.

```
NB:Top> ipx rip
NB:Top>Configure>IPX>RIP> ?
Available commands:
add             delete          list            listen
stat            top>            up>             ?
```

Commands at this level are used to configure and manage the IPX RIP. IPX RIP learns about routes to other indirect networks by receiving IPX routing table entries from other routers. The NetBlazer can participate in this process by transmitting its own IPX routing tables using the **ipx rip add** command.

ipx rip add <interface> [<interval (minutes)>]

The **ipx rip add** command starts RIP broadcasts on the network or interface. The NetBlazer is shipped with IPX RIP broadcasting turned off.

The <interface> parameter specifies the name of an interface over which IPX RIP broadcasts are sent periodically.

The optional <interval> parameter specifies the number of minutes between IPX routing table transmissions to the interface. The default is one minute. This is the recommended interval for a connection.

ipx rip delete <interface>

The **ipx rip delete** command stops broadcasting IPX RIP packets to an interface.

The <interface> parameter specifies which IPX interface should be removed from the IPX RIP interface list.

This command stops IPX RIP broadcasts to the interface.

ipx rip list

The **ipx rip list** command displays the list of interfaces to which the NetBlazer is sending routing information as well as the interfaces to which the NetBlazer is listening for routing information and requests.

ipx rip listen <interface>|all on|off

The **ipx rip listen** command tells the NetBlazer to put one or all current IPX interfaces in the listen state. You can also use the command to turn off the listen state for one or all interfaces. The NetBlazer is shipped with IPX RIP listening turned on.

ipx rip stat

The **ipx rip stat** command displays a table of IPX RIP statistics. The table shows the number of queries, responses, replies, broadcasts, send errors, and packets with incorrect length. It also shows a timer. Use this command only when debugging IPX RIP.

ipx route

The **ipx route** command takes you to the Configure IPX Route command level. To view a list of the commands available at this level, type a question mark (**?**) at the Configure IPX Route command level prompt.

```
NB:Top> ipx route
NB:Top>Configure>IPX>Route> ?
Available commands:
add             addprivate      bang            delete
flush           forget          learn           list
listlocal       ping            stat            top>
up>             ?
```

IPX route commands are used to maintain the NetBlazer IPX routing table.

When the IPX RIP server is running, RIP periodically updates the RIP acquired routes in the IPX routing table.

Use the **ipx route add** or **addprivate** commands to add static routes to the IPX routing table. Static routes that are designated as private are not advertised by RIP to other routers on the network. Also, the **ipx address** command adds static routes for Ethernet and Token Ring networks to the IPX routing table.

Use the **ipx route learn** command to get a snapshot of the IPX routes on a remote dial-up network. The **ipx route forget** command removes IPX routes acquired through the **ipx route learn** command.

ipx route add <dest-addr> <interface> [<gateway> [<metric>]]

The **ipx route add** command is used to add a static route to the IPX routing table.

The <dest-addr> parameter is the destination network (or host) IPX address.

The <interface> parameter specifies the name of the interface through which the packets are routed.

The optional <gateway> and <metric> parameters specify the IPX address of the gateway that the packets should be sent to for further routing and the number of hops required to reach the destination network.

To specify a route with a metric but no gateway, enter a gateway of **0** or **" "**.

Caution: *If you add a route with the same destination address as an existing route, the system generates an error message. Delete the old route before adding the new route.*

The following commands add three new static routes to the IPX routing table.

```
NB:Top>Configure>IPX>Route> add 410 Pearl
NB:Top>Configure>IPX>Route> add 350 Ruby "" 2
```

ipx route addprivate <dest-addr> <interface> [<gateway> [<metric>]]

The **ipx route addprivate** command is equivalent to the **ipx route add** command with one exception: Routes added to the IPX routing table through this command are not advertised to other routers by RIP.

ipx route bang <ipx-addr> [<length> [<count> [<interval> [<outstanding>]]]]

The **ipx route bang** command sends IPX echo requests to a specified remote NetBlazer. This command is similar to an **ipx route ping** or **ipx ping** command except **ipx route bang** prints only a single character (!) instead of a line of statistics for each response. Lost or out-of-sequence packets are shown as a dot (.). IPX messages are shown as an X in the **bang** display. Refer to the **ipx ping** command for an explanation of the parameters used by both commands.

ipx route delete <dest-addr> [<gateway>]

The **ipx route delete** command removes a specified route from the IPX routing table. The <dest-addr> parameter and the optional <gateway> parameter identify which route to remove.

ipx route flush

The **ipx route flush** command removes the IPX routing table entries that were created by RIP. This leaves only the static IPX routing table entries (those entered by hand). This command is useful immediately after restructuring a network so that the NetBlazer will not have to wait for the IPX routing table entries to expire (timeout) before learning new routes.

ipx route forget <interface>

The **ipx route forget** command is used to delete all learned routes for a specified interface.

The **ipx route forget** command is typically followed by use of the **ipx route learn** command to acquire new route information, and then a **save**.

ipx route learn <interface> [<seconds>]

The **ipx route learn** command, when given the name of the PPP interface, automatically establishes a connection (if not already open), and learns what networks exist on the far side.

This information obtained from the network is then automatically added to the routing table and specifically marked as having been learned. Learned route information is distinct from RIP acquired and manually added routes, thereby allowing learned routes to be automatically removed with the **forget** command.

ipx route list

The **ipx route list** command displays the current IPX routing table:

```
NB:Top>Configure>IPX>Route> list
Routing tables
Destination     Gateway          Flags  Refs     Use  Hops Ticks
Interface
321.*           330.E831         UG       0       0     1   37  topaz
331.*           330.E831         UG       0       0     1   37  topaz
334.*           334.80AD034C46   U        1       0     0    1  en0
```

Destination – Indicates the IPX address of the destination.

Gateway – Identifies any router or gateway that a packet should be sent to for further routing.

Flags – Indicates if a route is designated as shown in Table IPX-4.

Table IPX-4. Route Flags

Flag	Meaning	Flag	Meaning
A	Auto route	H	Route to a host
B	Backup route	P	Private Route
D	Dynamic	U	Route is up
G	Route to a gateway		

Refs – Indicates whether or not the route is in use now.

Use – Indicates the number of IPX packets that have used this route since the route was created.

Hops – Indicates the number of hops required to reach the destination or network.

Ticks - Indicates the number of ticks it takes for a packet to reach its destination. A tick is one-eighteenth (1/18) of a second.

Interface – Indicates the interface or device through which packets will be routed.

ipx route listlocal

The **ipx route listlocal** command displays a list of static routes that were not acquired through IPX RIP.

ipx route ping <ipx-addr> [<length> [<count> [<interval> [<outstanding>]]]]

The **ipx route ping** command sends IPX echo requests to a specified remote NetBlazer. This command is similar to an **ipx ping** command. Refer to the **ipx ping** command for an explanation of the parameters used by the **ipx route ping** command.

ipx route stat

The **ipx route stat** command displays a list of IPX routing statistics. The list shows the number of bad routing redirects, dynamically created routes, new gateways due to redirects, destinations found unreachable, and uses of a wildcard route. Use this command only when debugging an IPX route.

ipx sap

The **ipx sap** command takes you to the Configure IPX SAP command level. To view a list of the commands available at this level, type a question mark (**?**) at the Configure IPX SAP command level prompt.

```
NB:Top> ipx sap
NB:Top>Configure>IPX>SAP> ?
Available commands:
add             delete          list            listen
stat            table>          top>            up>
?
```

Commands at this level are used to configure and manage the IPX SAP. SAP learns about services by receiving service table entries from the servers.

The NetBlazer advertises its services to servers and it records in a service table the routes to the servers. Normally, SAP accepts service information from any server. You may not want the NetBlazer to accept service information from a particular server or servers.

ipx sap add <interface> [<interval (minutes)>]

The **ipx sap add** command starts SAP broadcasts on an interface. The NetBlazer is shipped with IPX SAP broadcasting turned off.

The <interface> parameter specifies the name of the interface.

The optional <interval> parameter specifies the number of minutes between SAP transmissions to the interface. The default is one minute. This is the recommended interval for a connection.

ipx sap delete <interface>

The **ipx sap delete** command stops SAP broadcasts on an interface.

The <interface> parameter specifies the name of the interface.

ipx sap list

The **ipx sap list** command displays a list of the names of interfaces to which the NetBlazer is sending service information. It also lists the names of interfaces to which the NetBlazer is listening for service information.

ipx sap listen <interface>|all on|off

The **ipx sap listen** command starts listening for SAP broadcasts on an interface.

You can also use the command to turn off the listen state for one or all interfaces. The NetBlazer is shipped with SAP listening turned on.

ipx sap stat

The **ipx sap stat** command displays SAP statistics. The statistics show the number of queries, responses, replies, broadcasts, send errors, table size, and packets with incorrect length or bad type. It also shows a timer count. Use this command only when debugging IPX SAP.

ipx sap table

The **ipx sap table** command takes you to the Configure IPX SAP Table command level. To view a list of the commands available at this level, type a question mark (**?**) at the Configure IPX SAP Table command level prompt.

```
NB:Top> ipx sap table
NB:Top>Configure>IPX>SAP>Table> ?
Available commands:
add             delete          list            top>
up>             ?
```

Commands at this level are used to add and delete service table entries.

ipx sap table add [<name> [<type> [<ipx-addr>]]]

The **ipx sap table add** command adds service table entries. The parameters discussed below are optional only in that you do not have to enter them when you enter the **sap table add** command. You can enter the command with no parameters. If you do so, the NetBlazer prompts you for the missing parameters.

The optional parameter <name> is the server name. The NetBlazer converts the name you enter to upper case.

The optional parameter <type> is the server type. Table IPX-5 lists some well known server types; numbers are hexadecimal.

The optional parameter <ipx-addr> is the server's IPX address in hexadecimal. The three parts of an IPX address are network, node, and socket. The socket number for a file server is 451h, for example. For some IPX socket numbers, see Table IPX-2 on page -3 in this section.

Table IPX-5. Server Types

Type	Meaning	Type	Meaning
FFFFh	Wild	0024h	Remote bridge server
0000h	Unknown	0027h	TCP/IP gateway
0003h	Print queue	002Dh	Time synchronization server
0004h	File server	0047h	Advertising print server
0005h	Job server	004Ah	Telebit ACS server
0006h	Gateway	007Ah	TES – NetWare for VMS
0007h	Print server	0098h	Netware access server
0009h	Archive server	009Eh	Portable NetWare
000Bh	Administration	0133h	NNS domain
0021h	NAS SNA gateway	8000h	Reserved up to

ipx sap table delete [<name> [<type>]]

The **ipx sap table delete** command deletes service table entries.

ipx sap table list

The **ipx sap table list** command displays the list of service table entries.

```
NB:Top>Configure>IPX>SAP>Table> list
Type Server Name                  Hops F Interface  Address
   4 PLUTO                           3 S topaz      00000321 000000000001
0451
```

Type – Indicates server type. See Table IPX-5 for server types.

Server Name – Lists the name you assign to the server.

F - Flag. **S** shows the entry is static; otherwise the column is blank.

Hops – Indicates the number of hops required to reach the server.

Interface – Indicates the interface or device through which the server is reached.

Address – Shows the server's IPX address in hexadecimal. See Table IPX-2 for IPX socket numbers.

ipx wan

The **ipx wan** command takes you to the Configure IPX WAN command level. To view a list of the commands available at this level, type a question mark (**?**) at the Configure IPX WAN command level prompt.

```
NB:Top> ipx wan
NB:Top>Configure>IPX>WAN> ?
Available commands:
add             delete          list            stat
top>            up>             ?
```

Commands at this level are used to enable or disable Novell's IPX Wide Area Network (WAN). IPX WAN can be used in place of Point-to-Point Protocol (PPP).

ipx wan add <interface>

The **ipx wan add** command enables IPX WAN on an interface.

ipx wan delete <interface>

The **ipx wan delete** command disables IPX WAN on an interface.

ipx wan list [<interface>]

The **ipx wan list** command lists WAN interfaces and their states. This command is usually used for debugging.

ipx wan stat

The **ipx wan stat** command displays IPX WAN statistics. The statistics cover the timer, state, retries, length, sequence, errors, type, and identification. It also shows a timer count. Use this command only when debugging IPX WAN.

Kerberos Commands

The **kerberos** command takes you to the Configure Kerberos command level. To view a list of the commands available at this level, type a question mark (?) at the Configure Kerberos command level.

```
NB:Top> kerberos
NB:Top>Configure>Kerberos> ?
Available commands:
gmtoffset       key>            list            realm
server>         translation>    top>            up>
?
```

Kerberos provides a way to authenticate users via an encrypted exchange over the network. Passwords are never sent as clear text over the network. Also, kerberos can be used to authenticate the user to the destination host in a telnet or rlogin session. This avoids sending the password as clear text over the network.

The user can update his password with the keberos server, so the system administrator is freed from the task of keeping the passwords current on the NetBlazer.

Also see the following Kerberos user commands in the Bin command descriptions: **kdestroy, kinit, klist,** and **kpasswd.**

kerberos gmtoffset [<hhmm>]

The **kerberos gmtoffset** command sets the offset in hours and minutes west of GMT. If no offset parameter is specified, the current offset is displayed.

kerberos key

The **kerberos key** command takes you to the Configure Kerberos Key command level. To view a list of the commands available at this level, type a question mark (?) at the Configure Kerberos Key command level.

```
NB:Top>Configure>Kerberos> key
NB:Top>Configure>Kerberos>Key> ?
Available commands:
add             delete          list            load
modify          save            top>            up>
?
```

These commands allow you to set up and maintain a list of private keys between the NetBlazer and the Kerberos Key Distribution Center (KDS).

kerberos key add [-k]

The **kerberos key add** command adsd a key to the Kerberos key list.

As illustrated below, this command prompts you for name, instance, realm, and key version number, asks for confirmation, and then asks for a password. The NetBlazer then converts the password to a key. If the -k option is used, the key is displayed.

```
NB:Top>Configure>Kerberos>Key> add -k
Name: netblazer
Instance: rcmd
Realm: TELEBIT.COM
Version number: 4
New principal: netblazer.rcmd@TELEBIT.COM; version 4
Is this correct? (ynq) [y]
Password:
Verifying, please re-enter Password:
Key: a7a8ae5e 1a648f40
Key successfully added.
Would you like to add another key? (y,n) [y] n
Remember to save your changes in the srvtab file.
```

kerberos key delete [-i]

The **kerberos key delete** command deletes all the keys. If the -i flag is given, the NetBlazer prompts for yes or no before deleting each key.

kerberos key list [-k]

The **kerberos key list** command lists the keys showing version number and principal name. If the -k option is given, the keys are shown also.

kerberos key load [<filename>]

The **kerberos key load** command loads the service keys from the specified srvtab file. The default filename is SRVTAB.

kerberos key modify [-i] [-k]

The **kerberos key modify** command changes all the keys by using the regular admin protocol. If the -i flag is given, the NetBlazer prompts for yes or no before changing each key. If the -k option is used, the old and new keys are displayed.

kerberos key save [<filename>]

The **kerberos key save** command saves the service keys to the specified srvtab file. The default filename is SRVTAB.

kerberos list

The **kerberos list** command displays the GMT offset, realm, and calls the list command from the Kerberos Key, Server, and Translation level.

kerberos realm [<realm>]

The **kerberos realm** command sets the local realm name. If you enter this command without specifying a realm name, the current realm setting is displayed.

```
NB:Top>Configure>Kerberos> realm TELEBIT.COM
NB:Top>Configure>Kerberos> realm
Local realm: TELEBIT.COM
```

Because Kerberos is case sensitive, you must match the case exactly. If a translation is not found for a host, it uses the domain name in all caps for the realm.

kerberos server

The **kerberos server** command takes you to the Configure Kerberos Server command level. To view a list of the commands available at this level, type a question mark (?) at the Configure Kerberos Server command level.

```
NB:Top>Configure>Kerberos> server
NB:Top>Configure>Kerberos>Server> ?
Available commands:
add             delete          list            top>
up>             ?
```

These commands allow you to set up and maintain a list of Kerberos servers that are accessible to the NetBlazer.

kerberos server add <server> [<realm>] [-admin]

The **kerberos server add** command adds a hostname and realm to the list of Kerberos servers that are accessible to the NetBlazer. If the <realm> is not specified, the local realm is assumed.

The -admin flag indicates that the host also provides an administrative database server. The NetBlazer requires access to at least one administrative server.

The following display illustrates how to add an administrative server to the NetBlazer's list of Kerberos servers.

```
NB:Top>Configure>Kerberos>Server> add napa TELEBIT.COM -admin
NB:Top>Configure>Kerberos>Server> list

Key distribution (and admin) server table:
Kerberos server                Realm                          Admin
napa                           TELEBIT.COM                    Yes
```

kerberos server delete <server>

The **kerberos server delete** command removes the specified server from the NetBlazer's list of Kerberos servers.

kerberos server list

The **kerberos server list** command displays a list of Kerberos servers that are accessible to the NetBlazer.

kerberos translation

The **kerberos translation** command takes you to the Configure Kerberos Translation command level. To view a list of the commands available at this level, type a question mark (?) at the Configure Kerberos Translation command level.

```
NB:Top>Configure>Kerberos> translation
NB:Top>Configure>Kerberos>Translation> ?
Available commands:
add             delete          list            top>
up>             ?
```

These commands allow you to set up and maintain a list of translations of hostnames to the Kerberos realm name.

kerberos translation add <hostname>|.<domainname> [<realm>]

The **kerberos server add** command adds a hostname or domain name to the translation list.

If a hostname exactly matches a <hostname>, the corresponding realm is the realm of the host. If a hostname does not match any <hostname>, but its domain exactly matches the <domainname>, the corresponding realm is the realm of the host. The <domainname> should be of the form .XXX.YYY, e.g. .TELEBIT.COM

If no translation entry applies, the host's realm is considered to be the hostname's domain portion converted to upper case.

kerberos translation delete <hostname>|.<domainname>

The **kerberos translation delete** command removes the specified hostname or domain name from the NetBlazer's list of translations.

kerberos translation list

The **kerberos translation list** command displays a list of Kerberos translations.

Msglog Command

The **msglog** [-<level>] <message> command is used to send a message to *syslogd* on the host system specified by the **syslog host** command. This command is useful for verifying that messages from the NetBlazer are reaching the appropriate log file on the *syslog* host system.

The -<level> parameter allows you to specific the message's severity level. If you do not specify a severity level, the message is sent as a level 6 (info) message.

The <message> parameter may be a single word or several words enclosed in quotes such as: "Test message from the NetBlazer."

If the message sent by **msglog** does not reach the host log file, refer to the **syslog host** command description for troubleshooting instructions.

Sessions Commands

The **sessions** command takes you to the Sessions command level. To view a list of the commands available at this level, type a question mark (?) at the Sessions command level.

```
NB:Top> session
NB:Top>Sessions> ?
Available commands:
abort           bang            binary          chatdial
close           connect         continue        dial
disconnect      eol             escape          fg
finger          ftp             jobs            kill
list            ntp             ping            rlogin
send            talk            telnet          terminal
tftp            top>            up>             ?
```

Commands at this level allow authorized users to start, stop, and control communication sessions with local or remote machines.

session abort [<session-number>]

The **session abort** command aborts an FTP data transfer operation. It is the only command permitted during an FTP session. If no <session-number> parameter is entered, the NetBlazer displays a list of active sessions and prompts you for the session number.

session bang <host> [<length> <count> <interval> <outstanding>]

The **session bang** command sends ICMP echo requests to the specified remote host. This command is equivalent to a **session ping** command except that instead of printing a line of statistics for each response, **session bang** only prints a single character (!). Lost or out of sequence packets are shown as a dot (.), and ICMP messages are shown as an X in the **bang** display. Refer to the **session ping** command for an explanation of the parameters used by both these commands.

session binary <session-number> [on|off]

The **session binary** command turns on or off the telnet binary option. This allows the NetBlazer to transmit and receive all 256 possible character codes. This is useful for such operations as X-terminals and DEC VT-200 series terminals. If binary mode is not enabled, the NetBlazer will transfer only a subset of the 7-bit ASCII code.

If the on or off parameter is omitted, the current telnet binary setting for the specified session number is displayed.

session chatdial <site-name>

The **session chatdial** <site-name> command establishes a character mode connection with a specified remote site and uses the chat script assigned to the site's dialout record to log in. The NetBlazer looks up the <site-name> in the dialout list to get the information needed to make the connection. Refer to the **dialout** command descriptions for instructions on setting up dial-out destinations.

session close [<session-number>]

The **session close** command closes the session specified by the <session-number> parameter. This is the preferred way to terminate a session. If no <session-number> is entered, the NetBlazer displays a list of active sessions and prompts you for the session number.

session connect <destination-name>|<destination-number>

The **session connect** command starts a telnet or dial-out session with a pre-defined destination. The NetBlazer first looks up the destination name in the user's assigned destination group to get the connection mode (telnet or dial) and address (domain name, IP address, or telephone number) and then makes the appropriate connection.

You can enter a destination number assigned to a destination in place of the destination name parameter. To get a list of destination numbers and names, press the return key when you are at the NetBlazer Sessions command level.

```
Top>Sessions> press the Return key
Accessible sites:
1  marketing          Marketing Computer
2  databank           Data Bank Service
```

The above example shows that two sites, marketing and databank, are accessible through the **connect** command for this user. Entering **connect 1** is equivalent to entering **connect marketing**. If you get an "All sites are accessible" message, you probably do not have a destination group assigned to your user ID account.

For more information about setting up destinations used by the **connect** command, see the **destination** command descriptions.

session continue [<session-number>]

The **session continue** command returns to the session specified by the <session-number> parameter. If no <session-number> parameter is entered, the NetBlazer prompts you for the session number.

You also can use the **session fg** command to continue a session.

session dial <site-name>|<phone-number>

The **session dial** <site-name> command establishes a character mode connection with a specified remote site. The NetBlazer looks up the <site-name> in the dialout list to get the information needed to make the connection. Refer to the **dialout** command descriptions for instructions on setting up dial-out destinations.

To allow dialing a telephone number directly, the system administrator must enter a **security numberdial enable** command.

session disconnect [<session-number>]

The **session disconnect** command terminates the session specified by the <session-number> parameter. It is intended for dial-out connections, but it can be used to close any session. If no <session-number> parameter is entered, the NetBlazer prompts you for the session number.

session eol [standard | unix]

The **session eol** command selects the proper end-of-line processing during a telnet session when the operator presses the return key. The default setting of "standard" should be used for most systems. The "unix" option should only be used when the remote system fails to recognize the carriage return key as indicating end-of-line.

session escape [<character>]

The **session escape** command displays or changes the keystroke used in a escape sequence to interrupt a session and get back to the command prompt. The escape sequence consists of the escape character followed by a carriage return.

Changing the escape character on the NetBlazer is desirable, for example, when telnet on your computer uses the same escape character as the NetBlazer. In this case, telnet traps the escape character that may have been intended for the NetBlazer. By changing the escape character on the NetBlazer to a value that does not match the telnet escape character, telnet allows the escape character to pass through to the NetBlazer.

Entering the command without the <character> parameter displays the current escape character, normally ctrl-].

The optional <character> parameter specifies which control character should be used as the escape character. Be sure to enter the desired keystroke exactly as you would use it. For example, if you decide to change the escape character to ctrl-/, you should hold down the control key and press the / key.

The escape character is set on a per session basis. The escape character used by one user may differ from the escape character used by other users logged into the system.

If a user with configuration privileges issues a **configure save** command after changing the escape character, that character becomes the default escape character when the NetBlazer is rebooted. The escape character setting for all other users only applies to their current login session.

session fg [<session-number>]

Like the **session continue** command, the **session fg** (foreground) command continues a session specified by the <session-number> parameter. If no <session-number> is entered, the NetBlazer prompts you for the session number.

session finger [<user>] [@<host>]

The **session finger** command invokes the finger client. The finger client permits a user to request information about a user at a host. The field <user> is a valid user ID on the system being queried. The field @<host> is the @ character immediately followed by an IP address or a domain name.

If <user> is specified without a host specification, the local (NetBlazer) finger server is queried about that user of the local NetBlazer. If just @<host> field is specified, the host being queried usually displays a list of possible users or a list of currently active users. This depends on the remote host and is not under the control of the NetBlazer.

Here are a couple of examples:

```
session finger julie@marketing.telebit.com
session finger @marketing.telebit.com
```

The first example returns information about the user julie on the host named marketing.telebit.com. The second example returns information about the current users at the host named marketing.telebit.com.

session ftp [<address>]

The **session ftp** command starts an FTP session with a host specified by the <address> parameter. The address is the host's IP address or domain name.

Once connected to the remote host, the standard set of FTP commands apply. These vary slightly from host to host so see the documentation for the particular remote host to which the connection is being established.

Caution: *Interrupting an FTP session while copying a file to the NetBlazer diskette might make the diskette unusable. Before starting the FTP session, make sure that you have a backup of your current system diskette. Do not interrupt the FTP session while it is copying a file to the NetBlazer diskette.*

session jobs

Like the **session list** command, the **session jobs** command displays a list of sessions currently running on the NetBlazer.

session kill [<session-number>]

The **session kill** command terminates the session specified by the <session-number> parameter without notifying the remote end. If no <session-number> is entered, the NetBlazer displays a list of active sessions and prompts you for the session number.

This command is used as a last attempt to close a session with a host that is no longer responding. Do not use this command to terminate a session with a remote host that is responding because the NetBlazer will terminate the session without telling the remote end that the connection is closed.

You should always first try to **close** a session and **kill** the session only if that fails.

session list

The **session list** command displays a list of sessions currently running on the NetBlazer. An example of a session status list is shown below:

```
NB:Top>Sessions> list
 #    S#  Type      Rcv-Q Snd-Q State        Remote socket      Pgroup
 0 $ 5   Telnet        0     0 Established  pearl (143.191.63.1:23)line17
```

The # field is the session number. An asterisk (*) next to the session number denotes the current session.

The S# field identifies the socket number. This field is use primarily for debugging purposes.

The Type field identifies the type of session; i.e., Telnet, FTP, Ping, etc.

The Rcv-Q displays the number of characters received on that session but not displayed. Continuing the session displays the stored data.

The Snd-Q displays the number of bytes transmitted but not acknowledged by the remote host.

The State field displays the current state of the TCP session. The most significant state is "Established" indicating that a connection exists with the remote host and that information may be transferred to/from that host. A state containing the word "SYN" indicates that the NetBlazer is attempting to establish a connection. A state containing the phrase "'FIN wait" or "Time wait" indicates that the session is being closed.

The Remote socket and Pgroup describe the session connection.

session ntp [-s] [-f] [-v] <ntp-host>

The **session ntp** command queries the specified NTP host for the current date and time.

Note: *Because the NetBlazer uses local time and NTP uses GMT or Universal time, you must set the offset from GMT using the* ***gmtoffset*** *command.*

Use the -s option to set the NetBlazer's time in sync with the NTP server.

If the NetBlazer's time is more than 1000 seconds out of sync with the NTP server, use the -f option to force the time setting.

The -v option displays more verbose information from the NTP server.

session ping <host> [<length> <count> <interval> <outstanding>]

The **session ping** command sends ICMP echo requests to the specified remote host. The NetBlazer then measures the amount of time required for the remote host to send a response.

The <host> parameter may be either a domain name or an IP address.

The optional <length> parameter is used to pad the request to a larger size specified in bytes. This parameter must be greater than 32 bytes (the minimum length allowed) to take effect. However, a value less than 32 can be used as placeholder in the command.

The optional <count> parameter specifies the number of ICMP requests to send to the remote host. If this parameter is not specified or 0 is entered as a placeholder, the NetBlazer continues issuing requests to the host until the ping session is stopped by pressing ctrl-C or by issuing a close, disconnect, or kill command.

The optional <interval> parameter specifies in seconds the time to wait between sending ICMP echo requests. This value may also be set to 0, and the NetBlazer will send pings with 10 ms separation (approximately 100 pings per second).

The optional <outstanding> parameter specifies the number of echo requests (pings) that may be transmitted before the NetBlazer stops and waits for their responses.

The ability to set the ping interval to 0 (10 ms between pings) and set the number of outstanding pings is a useful tool to place a significant load on a network without overloading the network.

session rlogin [-b|-8] [-noesc] [-l username] [<address>] [<port>]

The **session rlogin** command is used to establish an interactive session on remote host using current NetBlazer user ID as remote host login ID. If the remote host has an entry for NetBlazer in /etc/hosts.equiv, and the NetBlazer user ID matches the remote host user ID, the user is simply logged into remote host without requiring additional user name and password authentication on remote host.

The -b option starts a binary telnet session. This has the same result as escaping out of a rlogin session, turning on binary for that session, and then continuing the session.

The -8 option starts an rlogin session using 8-bit data in non-binary mode.

The -noesc option disables escape processing. This prevents the NetBlazer from enter command mode when an escape sequence is entered. Instead, the escape sequence is passed through to the remote end.

The -l option is used to overwrite the default user name. This is normally used to specify a different user name when logging onto a remote host that does not recognize the user's NetBlazer login.

Note: *The -noesc option may not be used if there are other sessions open for the same login (pgroup). Other sessions for the same login must be closed before an* ***rlogin -noesc*** *command will be accepted.*

The only way to return to the NetBlazer's command mode from an rlogin session that was started with the -noesc option is to close the session at the remote end (e.g., logout).

The optional <port> parameter is used to start a telnet with the host system using a port number other than telnet's well known port number 23.

session send <session-number> ao|ayt|brk|ec|el|esc|ga|ip|nop|synch

The **session send** command sends the special character sequences defined below to a remote host.

ao	Sends the telnet ao (abort output) sequence, which should cause the remote system to flush all output from the remote system to the user's terminal.
ayt	Sends the telnet ayt (are you there) sequence, to which the remote system may or may not choose to respond.
brk	Sends the telnet brk (break) sequence, which may have significance to the remote system.
ec	Sends the telnet ec (erase character) sequence, which should cause the remote system to erase the last character entered.
el	Sends the telnet el (erase line) sequence, which should cause the remote system to erase the line currently being entered.
esc	Sends the current telnet escape character. The CR or LF that would be expected by a NetBlazer as part of the escape sequence are not sent by this command; they may be sent by the user if required.
ga	Sends the telnet ga (go ahead) sequence, which likely has no significance to the remote system.
ip	Sends the telnet ip (interrupt process) sequence, which should cause the remote system to abort the currently running process.
nop	Sends the telnet nop (no operation) sequence.
synch	Sends the telnet synch sequence. This sequence causes the remote system to discard all previously typed but not yet read input. This sequence is sent as TCP urgent data and may not work if the remote system is a 4.2 BSD system – if it does not work, a lower case "r" may be echoed on the terminal.

session talk [<line-name>]

The **session talk** command is for debugging purposes. It allows you to connect to an asynchronous line and issue modem commands to an attached modem. The <line-name> parameter must be one of the lines listed by a **line list** command that is connected to a modem and not in use.

By default, only users with system configuration privileges can use the talk command. However, entering a **security user talk enable** allows all users to talk to a line.

To stop talking to the modem, send the escape character (control-]) to the NetBlazer and issue a **close**, **disconnect**, or **kill** command.

session telnet [-b|-8|-raw] [-noesc] [<address>] [<port>]

The **session telnet** command is used to establish a telnet session with a host specified by the <address> parameter. The address is the host's IP address or domain name.

The -b option starts a binary telnet session. This has the same result as escaping out of a telnet session, turning on binary for that session, and then continuing the session.

The -8 option starts a telnet session using 8-bit data in nonbinary mode.

The -raw option implies the -b and -noesc options and eliminates all telnet IAC, CR/LF/NUL, option, and suboption processing.

The -noesc option disables escape processing. This prevents the NetBlazer from enter command mode when an escape sequence is entered. Instead, the escape sequence is passed through to the remote end.

Note: *The -noesc option may not be used if other sessions are open for the same login (pgroup). Other sessions for the same login must be closed before a telnet -noesc command is accepted.*

The only way to return to the NetBlazer's command mode from a telnet session that was started with the -noesc option is to close the session at the remote end (e.g., logout).

The optional <port> parameter is used to start a telnet with the host system using a port number other than telnet's well-known port number 23.

session terminal [<terminal-type>] [-rows <rows>] [-cols <cols>]

The **session terminal** command sets or displays the terminal characteristics for the current login. Typically a telnet server running on a UNIX host uses this information to set the "TERM" environment variable. The desirable <terminal-type> values in that case are those listed in the host's /etc/termcap file or /etc/terminfo database.

Entering a session terminal command without any parameters displays the current login terminal characteristics.

```
NB:Top>Sessions> terminal
Terminal type: none
Rows: not set   Cols: not set
```

The next example shows you how to set the login terminal type to vt100.

```
NB:Top>Sessions>  terminal vt100
NB:Top>Sessions> terminal
Terminal type: "vt100"
Rows: not set   Cols: not set
```

To remove a terminal type parameter, use a **session terminal ""** command.

```
NB:Top>Sessions> terminal ""
NB:Top>Sessions> terminal
Terminal type: none
Rows: not set   Cols: not set
```

The -rows and -cols parameter defines the row and column characteristics of your terminal.

```
NB:Top>Sessions> terminal -rows 24 -cols 80
NB:Top>Sessions> terminal
Terminal type: none
Rows: 24   Cols: 80
```

session tftp [<address>]

The **session tftp** command starts an TFTP session with a host specified by the <address> parameter. The address is the host's IP address or domain name.

Once connected to the remote host, the standard set of TFTP commands apply. These vary slightly from host to host so see the documentation for the particular remote host to which the connection is being established.

Caution: *Interrupting an TFTP session while copying a file to the NetBlazer diskette might make the diskette unusable. Before starting the TFTP session, make sure that you have a backup of your current system diskette. Do not interrupt the TFTP session while it is copying a file to the NetBlazer diskette.*